ANEXOS

ANEXO I. DATASHEET ATMEGA2560

Features

- High Performance, Low Power AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
  - 135 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-Chip 2-cycle Multiplier
- Non-volatile Program and Data Memories
  - 64K/128K/256K Bytes of In-System Self-Programmable Flash
    Endurance: 10,000 Write/Erase Cycles
  - Optional Boot Code Section with Independent Lock Bits
    In-System Programming by On-chip Boot Program
    True Read-While-Write Operation
  - 4K Bytes EEPROM
    Endurance: 100,000 Write/Erase Cycles
  - 8K Bytes Internal SRAM
  - Up to 64K Bytes Optional External Memory Space
  - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - Four 16-bit Timer/Counter with Separate Prescaler, Compare- and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Four 8-bit PWM Channels
  - Six/Twelve PWM Channels with Programmable Resolution from 2 to 16 Bits (ATmega1281/2561, ATmega640/1280/2560)
  - Output Compare Modulator
  - 8/16-channel, 10-bit ADC
  - Two/Four Programmable Serial USART (ATmega1281/2561,ATmega640/1280/2560)
  - Master/Slave SPI Serial Interface
  - Byte Oriented 2-wire Serial Interface
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
  - 51/86 Programmable I/O Lines (ATmega1281/2561, ATmega640/1280/2560)
  - 64-lead (ATmega1281/2561)
  - 100-lead (ATmega640/1280/2560)
  - 100-lead TQFP (64-lead TQFP Option)
- Temperature Range:
  - -40°C to 85°C Industrial
- Speed Grade:
  - ATmega1281/2561V/ATmega640/1280/2560V:
    0 - 4 MHz @ 1.8 - 5.5V, 0 - 8 MHz @ 2.7 - 5.5V
  - ATmega640/1280/1281/2560/2561:
    0 - 8 MHz @ 2.7 - 5.5V, 0 - 16 MHz @ 4.5 - 5.5V

8-bit AVR® Microcontroller with 256K Bytes In-System Programmable Flash

ATmega1281/2561/V ATmega640/1280/2560/V

Advance Information

2549A–AVR–03/05

Pin Configurations

Figure 1. Pinout ATmega640/1280/2560

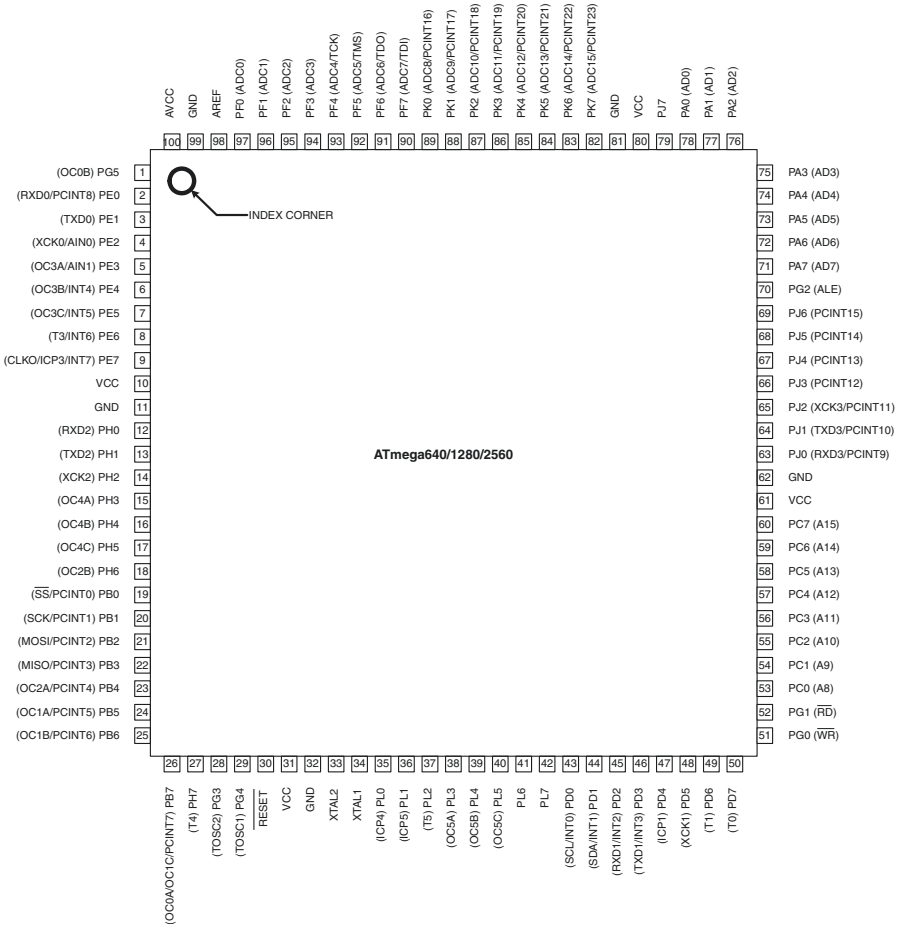
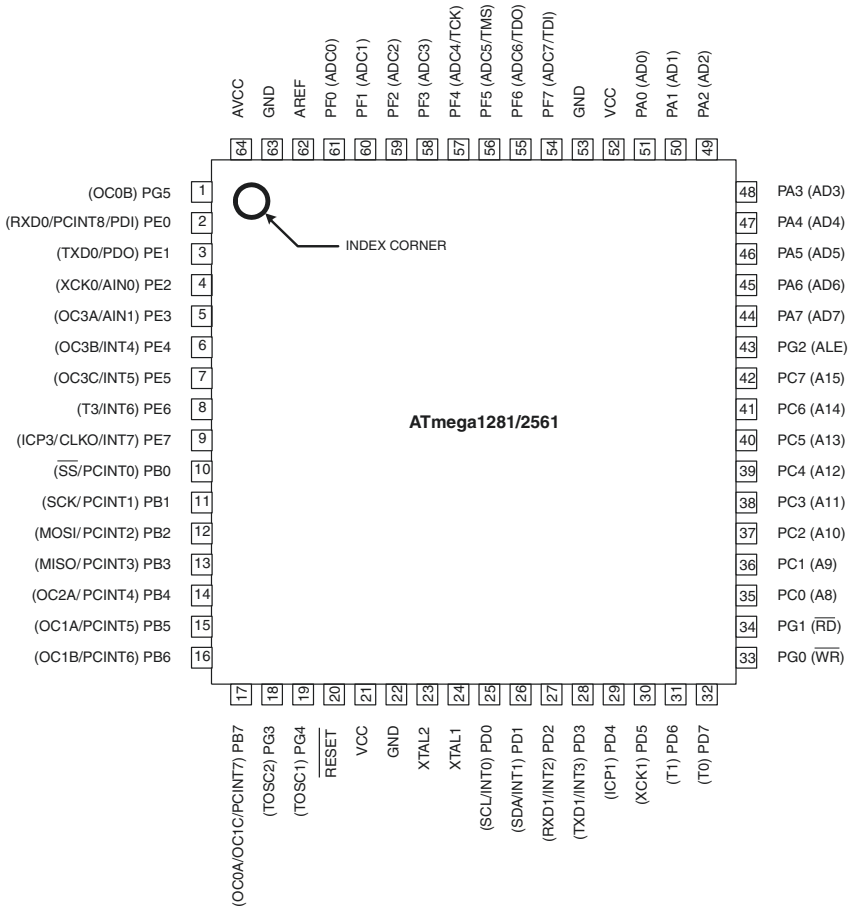


Figure 2. Pinout ATmega1281/2561



Note: The large center pad underneath the QFN/MLF package is made of metal and internally connected to GND. It should be soldered or glued to the board to ensure good mechanical stability. If the center pad is left unconnected, the package might loosen from the board.

Disclaimer

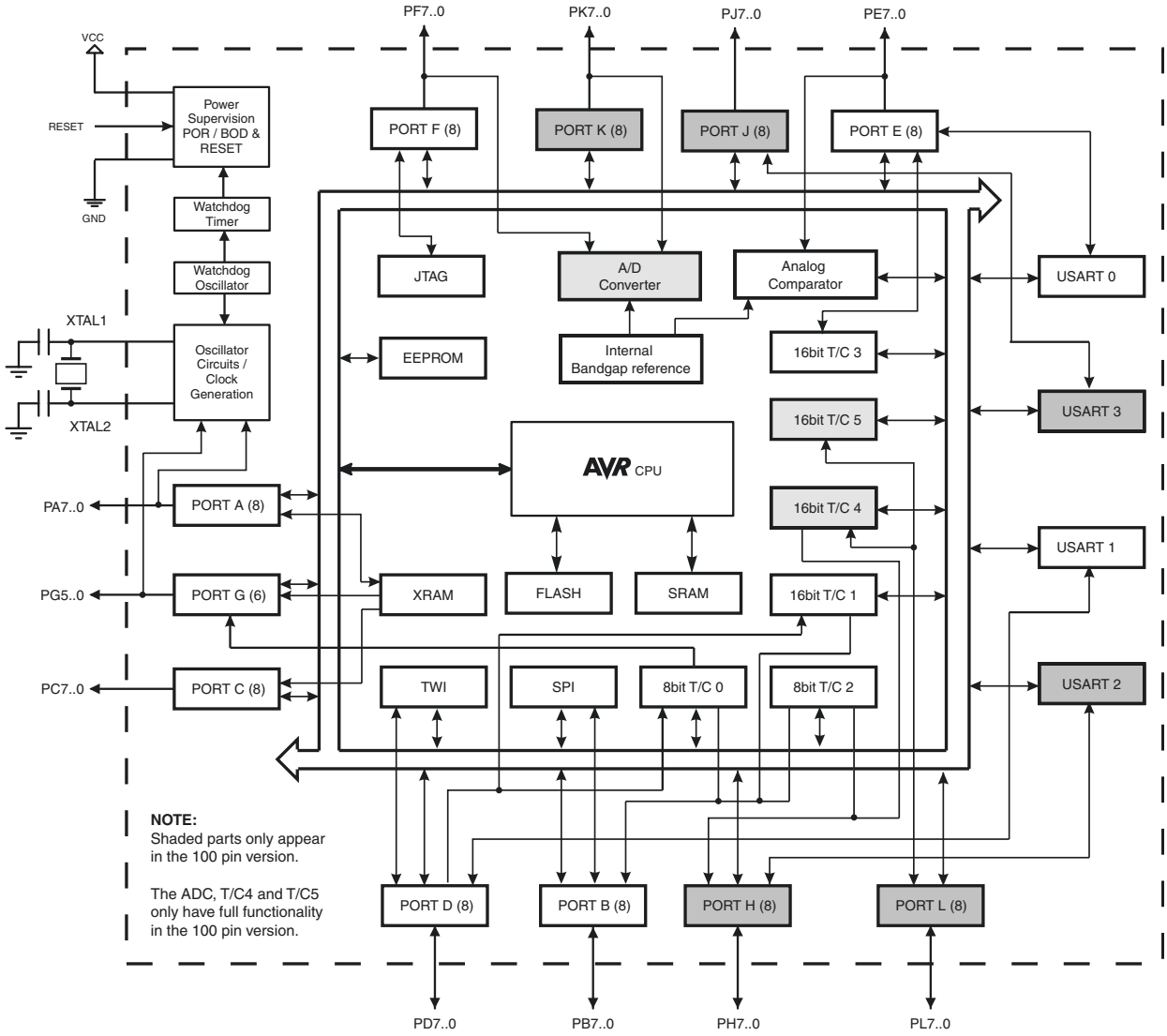
Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

Overview

The ATmega640/1280/1281/2560/2561 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega640/1280/1281/2560/2561 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 3. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega640/1280/1281/2560/2561 provides the following features: 64K/128K/256K bytes of In-System Programmable Flash with Read-While-Write capabilities, 4K bytes EEPROM, 8K bytes SRAM, 54/86 general purpose I/O lines, 32 general purpose working registers, Real Time Counter (RTC), six flexible Timer/Counters with compare modes and PWM, 4 USARTs, a byte oriented 2-wire Serial Interface, a 16-channel, 10-bit ADC with optional differential input stage with programmable gain, programmable Watchdog Timer with Internal Oscillator, an SPI serial port, IEEE std. 1149.1 compliant JTAG test interface, also used for accessing the On-chip Debug system and programming and six software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or Hardware Reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the Crystal/Resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using Atmel's high-density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega640/1280/1281/2560/2561 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega640/1280/1281/2560/2561 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

Comparison Between ATmega1281/2561 and ATmega640/1280/2560

Each device in the ATmega640/1280/1281/2560/2561 family differs only in memory size and number of pins. Table 1 summarizes the different configurations for the six devices.

Table 1. Configuration Summary

Device	Flash	EEPROM	RAM	General Purpose I/O pins	16 bits resolution PWM channels	Serial USARTs	ADC Channels
ATmega640	64KB	4KB	8KB	86	12	4	16
ATmega1280	128KB	4KB	8KB	86	12	4	16
ATmega1281	128KB	4KB	8KB	54	6	2	8
ATmega2560	256KB	4KB	8KB	86	12	4	16
ATmega2561	256KB	4KB	8KB	54	6	2	8

Pin Descriptions

VCC Digital supply voltage.

GND Ground.

Port A (PA7..PA0) Port A is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port A pins that are externally pulled low will source current if the pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port A also serves the functions of various special features of the ATmega640/1280/1281/2560/2561 as listed on page 88.

Port B (PB7..PB0) Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B has better driving capabilities than the other ports.

Port B also serves the functions of various special features of the ATmega640/1280/1281/2560/2561 as listed on page 89.

Port C (PC7..PC0) Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port C also serves the functions of special features of the ATmega640/1280/1281/2560/2561 as listed on page 92.

Port D (PD7..PD0) Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source

current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D also serves the functions of various special features of the ATmega640/1280/1281/2560/2561 as listed on page 94.

Port E (PE7..PE0)

Port E is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port E output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port E pins that are externally pulled low will source current if the pull-up resistors are activated. The Port E pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port E also serves the functions of various special features of the ATmega640/1280/1281/2560/2561 as listed on page 96.

Port F (PF7..PF0)

Port F serves as analog inputs to the A/D Converter.

Port F also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port F output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port F pins that are externally pulled low will source current if the pull-up resistors are activated. The Port F pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PF7(TDI), PF5(TMS), and PF4(TCK) will be activated even if a reset occurs.

Port F also serves the functions of the JTAG interface.

Port G (PG5..PG0)

Port G is a 6-bit I/O port with internal pull-up resistors (selected for each bit). The Port G output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port G pins that are externally pulled low will source current if the pull-up resistors are activated. The Port G pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port G also serves the functions of various special features of the ATmega640/1280/1281/2560/2561 as listed on page 102.

Port H (PH7..PH0)

Port H is a 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port H output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port H pins that are externally pulled low will source current if the pull-up resistors are activated. The Port H pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port H also serves the functions of various special features of the ATmega640/1280/1281/2560/2561 as listed on page 104.

Port J (PJ7..PJ0)

Port J is a 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port J output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port J pins that are externally pulled low will source current if the pull-up resistors are activated. The Port J pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port J also serves the functions of various special features of the ATmega640/1280/1281/2560/2561 as listed on page 106.

Port K (PK7..PK0)

Port K serves as analog inputs to the A/D Converter.

Port K is a 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port K output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port K pins that are externally pulled low will source current if the pull-up resistors are activated. The Port K pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port K also serves the functions of various special features of the ATmega640/1280/1281/2560/2561 as listed on page 108.

Port L (PL7..PL0)

Port L is a 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port L output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port L pins that are externally pulled low will source current if the pull-up resistors are activated. The Port L pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port L also serves the functions of various special features of the ATmega640/1280/1281/2560/2561 as listed on page 110.

$\overline{\text{RESET}}$

Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 23 on page 58. Shorter pulses are not guaranteed to generate a reset.

XTAL1

Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting Oscillator amplifier.

AVCC

AVCC is the supply voltage pin for Port F and the A/D Converter. It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter.

AREF

This is the analog reference pin for the A/D Converter.

About Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C compiler documentation for more details.

These code examples assume that the part specific header file is included before compilation. For I/O registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instructions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBRS", "SBRC", "SBR", and "CBR".

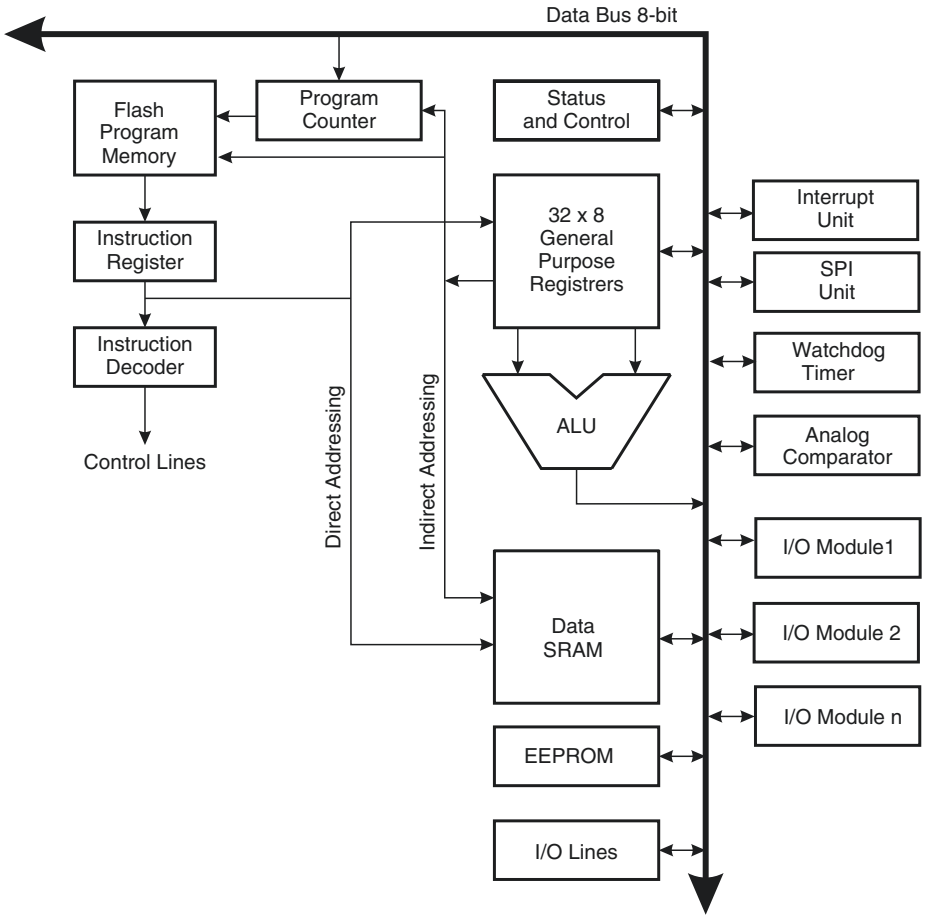
AVR CPU Core

Introduction

This section discusses the AVR core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

Architectural Overview

Figure 4. Block Diagram of the AVR Architecture



In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is In-System Reprogrammable Flash memory.

The fast-access Register File contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two operands are output from the Register File,

the operation is executed, and the result is stored back in the Register File – in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing – enabling efficient address calculations. One of the these address pointers can also be used as an address pointer for look up tables in Flash program memory. These added function registers are the 16-bit X-, Y-, and Z-register, described later in this section.

The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed in the ALU. After an arithmetic operation, the Status Register is updated to reflect information about the result of the operation.

Program flow is provided by conditional and unconditional jump and call instructions, able to directly address the whole address space. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

Program Flash memory space is divided in two sections, the Boot Program section and the Application Program section. Both sections have dedicated Lock bits for write and read/write protection. The SPM instruction that writes into the Application Flash memory section must reside in the Boot Program section.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the Stack. The Stack is effectively allocated in the general data SRAM, and consequently the Stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the SP in the Reset routine (before subroutines or interrupts are executed). The Stack Pointer (SP) is read/write accessible in the I/O space. The data SRAM can easily be accessed through the five different addressing modes supported in the AVR architecture.

The memory spaces in the AVR architecture are all linear and regular memory maps.

A flexible interrupt module has its control registers in the I/O space with an additional Global Interrupt Enable bit in the Status Register. All interrupts have a separate Interrupt Vector in the Interrupt Vector table. The interrupts have priority in accordance with their Interrupt Vector position. The lower the Interrupt Vector address, the higher the priority.

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, SPI, and other I/O functions. The I/O Memory can be accessed directly, or as the Data Space locations following those of the Register File, 0x20 - 0x5F. In addition, the ATmega640/1280/1281/2560/2561 has Extended I/O space from 0x60 - 0x1FF in SRAM where only the ST/STS/STD and LD/LDS/LDD instructions can be used.

ALU – Arithmetic Logic Unit

The high-performance AVR ALU operates in direct connection with all the 32 general purpose working registers. Within a single clock cycle, arithmetic operations between general purpose registers or between a register and an immediate are executed. The ALU operations are divided into three main categories – arithmetic, logical, and bit-functions. Some implementations of the architecture also provide a powerful multiplier supporting both signed/unsigned multiplication and fractional format. See the “Instruction Set” section for a detailed description.

Status Register

The Status Register contains information about the result of the most recently executed arithmetic instruction. This information can be used for altering program flow in order to perform conditional operations. Note that the Status Register is updated after all ALU operations, as specified in the Instruction Set Reference. This will in many cases

remove the need for using the dedicated compare instructions, resulting in faster and more compact code.

The Status Register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt. This must be handled by software.

The AVR Status Register – SREG – is defined as:

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – I: Global Interrupt Enable**

The Global Interrupt Enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the Global Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

- **Bit 6 – T: Bit Copy Storage**

The Bit Copy instructions BLD (Bit LoaD) and BST (Bit STore) use the T-bit as source or destination for the operated bit. A bit from a register in the Register File can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the Register File by the BLD instruction.

- **Bit 5 – H: Half Carry Flag**

The Half Carry Flag H indicates a Half Carry in some arithmetic operations. Half Carry Is useful in BCD arithmetic. See the “Instruction Set Description” for detailed information.

- **Bit 4 – S: Sign Bit, S = N $\oplus$ V**

The S-bit is always an exclusive or between the Negative Flag N and the Two’s Complement Overflow Flag V. See the “Instruction Set Description” for detailed information.

- **Bit 3 – V: Two’s Complement Overflow Flag**

The Two’s Complement Overflow Flag V supports two’s complement arithmetics. See the “Instruction Set Description” for detailed information.

- **Bit 2 – N: Negative Flag**

The Negative Flag N indicates a negative result in an arithmetic or logic operation. See the “Instruction Set Description” for detailed information.

- **Bit 1 – Z: Zero Flag**

The Zero Flag Z indicates a zero result in an arithmetic or logic operation. See the “Instruction Set Description” for detailed information.

- **Bit 0 – C: Carry Flag**

The Carry Flag C indicates a carry in an arithmetic or logic operation. See the “Instruction Set Description” for detailed information.

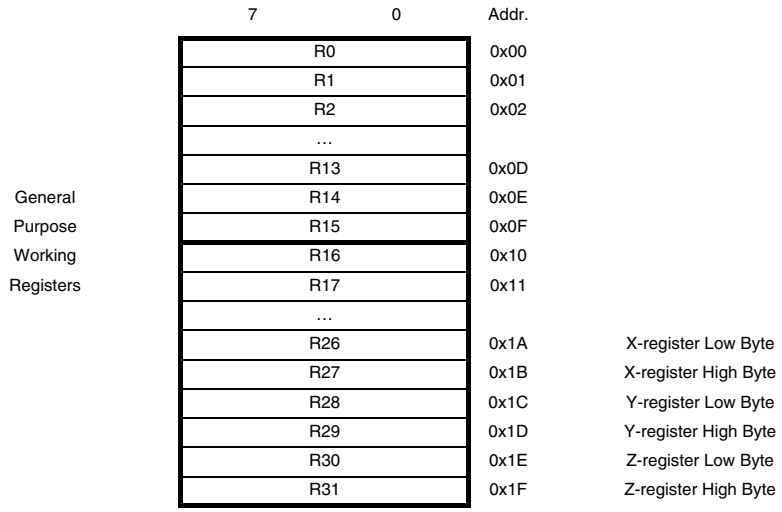
General Purpose Register File

The Register File is optimized for the AVR Enhanced RISC instruction set. In order to achieve the required performance and flexibility, the following input/output schemes are supported by the Register File:

- One 8-bit output operand and one 8-bit result input
- Two 8-bit output operands and one 8-bit result input
- Two 8-bit output operands and one 16-bit result input
- One 16-bit output operand and one 16-bit result input

Figure 5 shows the structure of the 32 general purpose working registers in the CPU.

Figure 5. AVR CPU General Purpose Working Registers



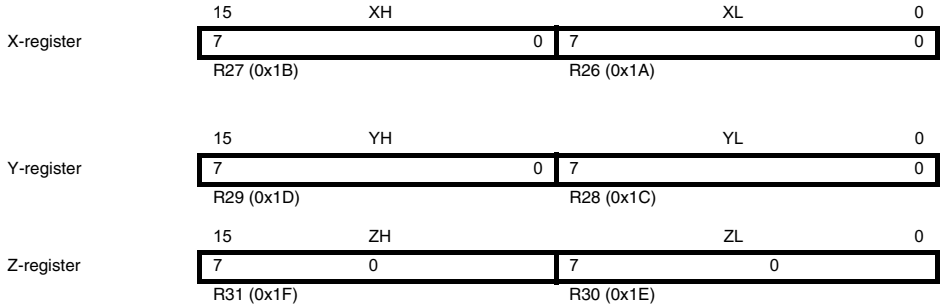
Most of the instructions operating on the Register File have direct access to all registers, and most of them are single cycle instructions.

As shown in Figure 5, each register is also assigned a data memory address, mapping them directly into the first 32 locations of the user Data Space. Although not being physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X-, Y- and Z-pointer registers can be set to index any register in the file.

The X-register, Y-register, and Z-register

The registers R26..R31 have some added functions to their general purpose usage. These registers are 16-bit address pointers for indirect addressing of the data space. The three indirect address registers X, Y, and Z are defined as described in Figure 6.

Figure 6. The X-, Y-, and Z-registers



In the different addressing modes these address registers have functions as fixed displacement, automatic increment, and automatic decrement (see the instruction set reference for details).

Stack Pointer

The Stack is mainly used for storing temporary data, for storing local variables and for storing return addresses after interrupts and subroutine calls. The Stack Pointer Register always points to the top of the Stack. Note that the Stack is implemented as growing from higher memory locations to lower memory locations. This implies that a Stack PUSH command decreases the Stack Pointer.

The Stack Pointer points to the data SRAM Stack area where the Subroutine and Interrupt Stacks are located. This Stack space in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. The Stack Pointer must be set to point above 0x0200. The initial value of the stack pointer is the last address of the internal SRAM. The Stack Pointer is decremented by one when data is pushed onto the Stack with the PUSH instruction, and it is decremented by three when the return address is pushed onto the Stack with subroutine call or interrupt. The Stack Pointer is incremented by one when data is popped from the Stack with the POP instruction, and it is incremented by three when data is popped from the Stack with return from subroutine RET or return from interrupt RETI.

The AVR Stack Pointer is implemented as two 8-bit registers in the I/O space. The number of bits actually used is implementation dependent. Note that the data space in some implementations of the AVR architecture is so small that only SPL is needed. In this case, the SPH Register will not be present.

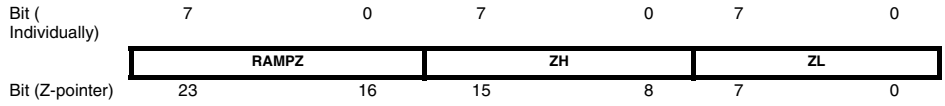
Bit	15	14	13	12	11	10	9	8	
	SP15	**SP14**	**SP13**	**SP12**	**SP11**	**SP10**	**SP9**	**SP8**	**SPH**
	SP7	**SP6**	**SP5**	**SP4**	**SP3**	**SP2**	**SP1**	**SP0**	**SPL**
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	1	
	1	1	1	1	1	1	1	1	

Extended Z-pointer Register for ELPM/SPM - RAMPZ

Bit	7	6	5	4	3	2	1	0	
	RAMPZ7	RAMPZ6	RAMPZ5	RAMPZ4	RAMPZ3	RAMPZ2	RAMPZ1	RAMPZ0	RAMPZ
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

For ELPM/SPM instructions, the Z-pointer is a concatenation of RAMPZ, ZH, and ZL, as shown in Figure 7. Note that LPM is not affected by the RAMPZ setting.

Figure 7. The Z-pointer used by ELPM and SPM



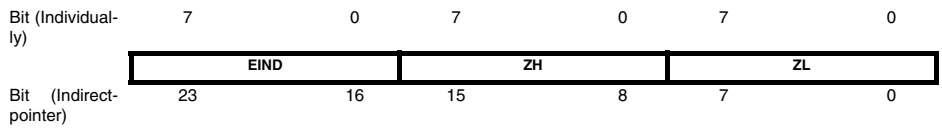
The actual number of bits is implementation dependent. Unused bits in an implementation will always read as zero. For compatibility with future devices, be sure to write these bits to zero.

Extended Indirect Register - EIND

Bit	7	6	5	4	3	2	1	0	
	EIND7	EIND6	EIND5	EIND4	EIND3	EIND2	EIND1	EIND0	EIND
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

For EICALL/EIJMP instructions, the Indirect-pointer to the subroutine/routine is a concatenation of EIND, ZH, and ZL, as shown in Figure 8. Note that ICALL and IJMP are not affected by the EIND setting.

Figure 8. The Indirect-pointer used by EICALL and EIJMP



The actual number of bits is implementation dependent. Unused bits in an implementation will always read as zero. For compatibility with future devices, be sure to write these bits to zero.

Instruction Execution Timing

This section describes the general access timing concepts for instruction execution. The AVR CPU is driven by the CPU clock clk_{CPU}, directly generated from the selected clock source for the chip. No internal clock division is used.

Figure 9 shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast-access Register File concept. This is the basic pipelining concept to obtain up to 1 MIPS per MHz with the corresponding unique results for functions per cost, functions per clocks, and functions per power-unit.

Figure 9. The Parallel Instruction Fetches and Instruction Executions

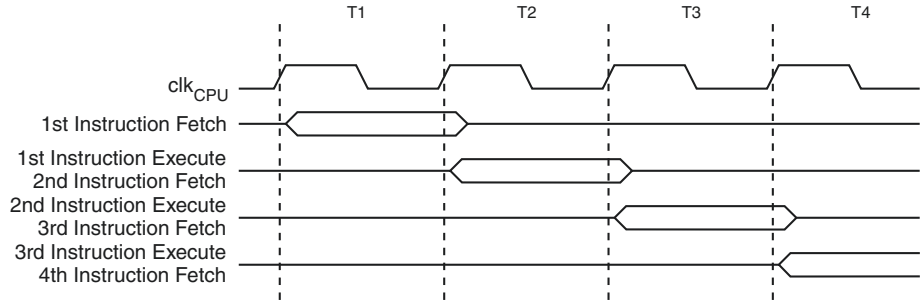
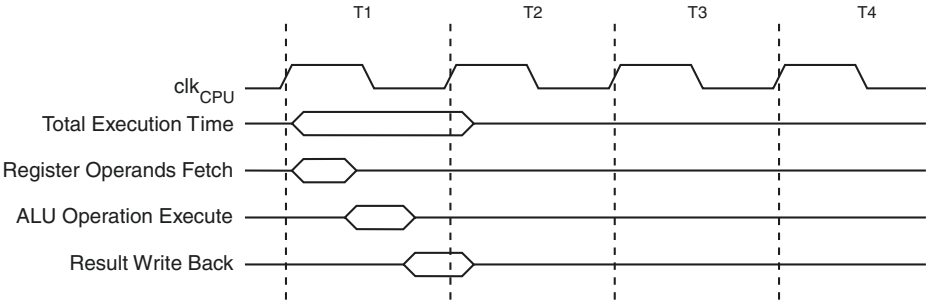


Figure 10 shows the internal timing concept for the Register File. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

Figure 10. Single Cycle ALU Operation



Reset and Interrupt Handling

The AVR provides several different interrupt sources. These interrupts and the separate Reset Vector each have a separate program vector in the program memory space. All interrupts are assigned individual enable bits which must be written logic one together with the Global Interrupt Enable bit in the Status Register in order to enable the interrupt. Depending on the Program Counter value, interrupts may be automatically disabled when Boot Lock bits BLB02 or BLB12 are programmed. This feature improves software security. See the section “Memory Programming” on page 335 for details.

The lowest addresses in the program memory space are by default defined as the Reset and Interrupt Vectors. The complete list of vectors is shown in “Interrupts” on page 69. The list also determines the priority levels of the different interrupts. The lower the address the higher is the priority level. RESET has the highest priority, and next is INT0 – the External Interrupt Request 0. The Interrupt Vectors can be moved to the start of the Boot Flash section by setting the IVSEL bit in the MCU Control Register (MCUCR). Refer to “Interrupts” on page 69 for more information. The Reset Vector can also be moved to the start of the Boot Flash section by programming the BOOTRST Fuse, see “Memory Programming” on page 335.

When an interrupt occurs, the Global Interrupt Enable I-bit is cleared and all interrupts are disabled. The user software can write logic one to the I-bit to enable nested interrupts. All enabled interrupts can then interrupt the current interrupt routine. The I-bit is automatically set when a Return from Interrupt instruction – RETI – is executed.

There are basically two types of interrupts. The first type is triggered by an event that sets the Interrupt Flag. For these interrupts, the Program Counter is vectored to the actual Interrupt Vector in order to execute the interrupt handling routine, and hardware clears the corresponding Interrupt Flag. Interrupt Flags can also be cleared by writing a logic one to the flag bit position(s) to be cleared. If an interrupt condition occurs while the corresponding interrupt enable bit is cleared, the Interrupt Flag will be set and remembered until the interrupt is enabled, or the flag is cleared by software. Similarly, if one or more interrupt conditions occur while the Global Interrupt Enable bit is cleared, the corresponding Interrupt Flag(s) will be set and remembered until the Global Interrupt Enable bit is set, and will then be executed by order of priority.

The second type of interrupts will trigger as long as the interrupt condition is present. These interrupts do not necessarily have Interrupt Flags. If the interrupt condition disappears before the interrupt is enabled, the interrupt will not be triggered.

When the AVR exits from an interrupt, it will always return to the main program and execute one more instruction before any pending interrupt is served.

Note that the Status Register is not automatically stored when entering an interrupt routine, nor restored when returning from an interrupt routine. This must be handled by software.

When using the CLI instruction to disable interrupts, the interrupts will be immediately disabled. No interrupt will be executed after the CLI instruction, even if it occurs simultaneously with the CLI instruction. The following example shows how this can be used to avoid interrupts during the timed EEPROM write sequence..

Assembly Code Example

```
in  r16, SREG      ; store SREG value
cli     ; disable interrupts during timed sequence
sbi EECR, EEMPE    ; start EEPROM write
sbi EECR, EEPE
out SREG, r16      ; restore SREG value (I-bit)
```

C Code Example

```
char cSREG;
cSREG = SREG;  /* store SREG value */
/* disable interrupts during timed sequence */
__disable_interrupt();
EECR |= (1<<EEMPE); /* start EEPROM write */
EECR |= (1<<EEPE);
SREG = cSREG; /* restore SREG value (I-bit) */
```

When using the SEI instruction to enable interrupts, the instruction following SEI will be executed before any pending interrupts, as shown in this example.

Assembly Code Example

```
sei   ; set Global Interrupt Enable
sleep ; enter sleep, waiting for interrupt
; note: will enter sleep before any pending
; interrupt(s)
```

C Code Example

```
__enable_interrupt(); /* set Global Interrupt Enable */
__sleep(); /* enter sleep, waiting for interrupt */
/* note: will enter sleep before any pending interrupt(s) */
```

Interrupt Response Time

The interrupt execution response for all the enabled AVR interrupts is five clock cycles minimum. After five clock cycles the program vector address for the actual interrupt handling routine is executed. During these five clock cycle period, the Program Counter is pushed onto the Stack. The vector is normally a jump to the interrupt routine, and this jump takes three clock cycles. If an interrupt occurs during execution of a multi-cycle instruction, this instruction is completed before the interrupt is served. If an interrupt occurs when the MCU is in sleep mode, the interrupt execution response time is increased by five clock cycles. This increase comes in addition to the start-up time from the selected sleep mode.

A return from an interrupt handling routine takes five clock cycles. During these five clock cycles, the Program Counter (three bytes) is popped back from the Stack, the Stack Pointer is incremented by three, and the I-bit in SREG is set.

ANEXO II. DATASHEET ADXL345

3-Axis, ±2 *g*/±4 *g*/±8 *g*/±16 *g* Digital Accelerometer

Data Sheet **ADXL345**

FEATURES

- Ultralow power: as low as 23 µA in measurement mode and 0.1 µA in standby mode at V_S = 2.5 V (typical)
- Power consumption scales automatically with bandwidth
- User-selectable resolution
  - Fixed 10-bit resolution
  - Full resolution, where resolution increases with *g* range, up to 13-bit resolution at ±16 *g* (maintaining 4 mg/LSB scale factor in all *g* ranges)
- Embedded memory management system with FIFO technology minimizes host processor load
- Single tap/double tap detection
- Activity/inactivity monitoring
- Free-fall detection
- Supply voltage range: 2.0 V to 3.6 V
- I/O voltage range: 1.7 V to V_S
- SPI (3- and 4-wire) and I^2C digital interfaces
- Flexible interrupt modes mappable to either interrupt pin
- Measurement ranges selectable via serial command
- Bandwidth selectable via serial command
- Wide temperature range (−40°C to +85°C)
- 10,000 *g* shock survival
- Pb free/RoHS compliant
- Small and thin: 3 mm × 5 mm × 1 mm LGA package

APPLICATIONS

- Handsets
- Medical instrumentation
- Gaming and pointing devices
- Industrial instrumentation
- Personal navigation devices
- Hard disk drive (HDD) protection

GENERAL DESCRIPTION

The ADXL345 is a small, thin, ultralow power, 3-axis accelerometer with high resolution (13-bit) measurement at up to ±16 *g*. Digital output data is formatted as 16-bit twos complement and is accessible through either a SPI (3- or 4-wire) or I^2C digital interface.

The ADXL345 is well suited for mobile device applications. It measures the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion or shock. Its high resolution (3.9 mg/LSB) enables measurement of inclination changes less than 1.0°.

Several special sensing functions are provided. Activity and inactivity sensing detect the presence or lack of motion by comparing the acceleration on any axis with user-set thresholds. Tap sensing detects single and double taps in any direction. Free-fall sensing detects if the device is falling. These functions can be mapped individually to either of two interrupt output pins. An integrated memory management system with a 32-level first in, first out (FIFO) buffer can be used to store data to minimize host processor activity and lower overall system power consumption.

Low power modes enable intelligent motion-based power management with threshold sensing and active acceleration measurement at extremely low power dissipation.

The ADXL345 is supplied in a small, thin, 3 mm × 5 mm × 1 mm, 14-lead, plastic package.

FUNCTIONAL BLOCK DIAGRAM

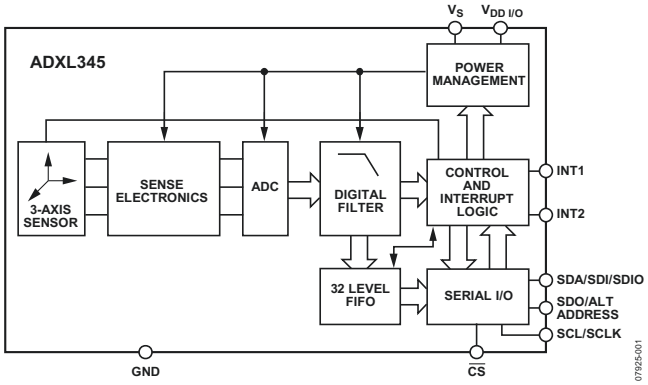


Figure 1.

Rev. E Document Feedback

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700
Technical Support www.analog.com

TABLE OF CONTENTS

REVISION HISTORY

6/15—Rev. D to Rev. E

Changes to Features Section and General Description Section....1
Change to Figure 36....15
Change to FIFO Section....21

2/13—Rev. C to Rev. D

Changes to Figure 13, Figure 14, and Figure 15....9
Change to Table 15....22

5/11—Rev. B to Rev. C

Added Preventing Bus Traffic Errors Section....15
Changes to Figure 37, Figure 38, Figure 39....16
Changes to Table 12....19
Changes to Using Self-Test Section....31
Changes to Axes of Acceleration Sensitivity Section....35

11/10—Rev. A to Rev. B

Change to 0 *g* Offset vs. Temperature for Z-Axis Parameter, Table 1....4
Changes to Figure 10 to Figure 15....9
Changes to Ordering Guide....37

4/10—Rev. 0 to Rev. A

Changes to Features Section and General Description Section....1
Changes to Specifications Section....3
Changes to Table 2 and Table 3....5
Added Package Information Section, Figure 2, and Table 4; Renumbered Sequentially....5
Changes to Pin 12 Description, Table 5....6
Added Typical Performance Characteristics Section....7
Changes to Theory of Operation Section and Power Sequencing Section....12
Changes to Powers Savings Section, Table 7, Table 8, Auto Sleep Mode Section, and Standby Mode Section....13
Changes to SPI Section....14
Changes to Figure 36 to Figure 38....15
Changes to Table 9 and Table 10....16
Changes to I^2C Section and Table 11....17
Changes to Table 12....18
Changes to Interrupts Section, Activity Section, Inactivity Section, and FREE_FALL Section....19
Added Table 13....19
Changes to FIFO Section....20
Changes to Self-Test Section and Table 15 to Table 18....21
Added Figures 42 and Table 14....21
Changes to Table 19....22
Changes to Register 0x1D—THRESH_TAP (Read/Write) Section, Register 0x1E, Register 0x1F, Register 0x20—OFSX, OFSY, OSXZ (Read/Write) Section, Register 0x21—DUR (Read/Write) Section, Register 0x22—Latent (Read/Write) Section, and Register 0x23—Window (Read/Write) Section....23
Changes to ACT_X Enable Bits and INACT_X Enable Bit Section, Register 0x28—THRESH_FF (Read/Write) Section, Register 0x29—TIME_FF (Read/Write) Section, Asleep Bit Section, and AUTO_SLEEP Bit Section....24
Changes to Sleep Bit Section....25
Changes to Power Supply Decoupling Section, Mechanical Considerations for Mounting Section, and Tap Detection Section....27
Changes to Threshold Section....28
Changes to Sleep Mode vs. Low Power Mode Section....29
Added Offset Calibration Section....29
Changes to Using Self-Test Section....30
Added Data Formatting of Upper Data Rates Section, Figure 48, and Figure 49....31
Added Noise Performance Section, Figure 50 to Figure 52, and Operation at Voltages Other Than 2.5 V Section....32
Added Offset Performance at Lowest Data Rates Section and Figure 53 to Figure 55....33

6/09—Revision 0: Initial Version

SPECIFICATIONS

T_A = 25°C, V_S = 2.5 V, $V_{DD\ I/O}$ = 1.8 V, acceleration = 0 *g*, C_S = 10 µF tantalum, $C_{I/O}$ = 0.1 µF, output data rate (ODR) = 800 Hz, unless otherwise noted. All minimum and maximum specifications are guaranteed. Typical specifications are not guaranteed.

Table 1.

Parameter	Test Conditions	Min	Typ[1]	Max	Unit
SENSOR INPUT	Each axis				
Measurement Range	User selectable		±2, ±4, ±8, ±16		*g*
Nonlinearity	Percentage of full scale		±0.5		%
Inter-Axis Alignment Error			±0.1		Degrees
Cross-Axis Sensitivity[2]			±1		%
OUTPUT RESOLUTION	Each axis				
All *g* Ranges	10-bit resolution		10		Bits
±2 *g* Range	Full resolution		10		Bits
±4 *g* Range	Full resolution		11		Bits
±8 *g* Range	Full resolution		12		Bits
±16 *g* Range	Full resolution		13		Bits
SENSITIVITY	Each axis				
Sensitivity at X_{OUT}, Y_{OUT}, Z_{OUT}	All *g*-ranges, full resolution	230	256	282	LSB/*g*
	±2 *g*, 10-bit resolution	230	256	282	LSB/*g*
	±4 *g*, 10-bit resolution	115	128	141	LSB/*g*
	±8 *g*, 10-bit resolution	57	64	71	LSB/*g*
	±16 *g*, 10-bit resolution	29	32	35	LSB/*g*
Sensitivity Deviation from Ideal	All *g*-ranges		±1.0		%
Scale Factor at X_{OUT}, Y_{OUT}, Z_{OUT}	All *g*-ranges, full resolution	3.5	3.9	4.3	m*g*/LSB
	±2 *g*, 10-bit resolution	3.5	3.9	4.3	m*g*/LSB
	±4 *g*, 10-bit resolution	7.1	7.8	8.7	m*g*/LSB
	±8 *g*, 10-bit resolution	14.1	15.6	17.5	m*g*/LSB
	±16 *g*, 10-bit resolution	28.6	31.2	34.5	m*g*/LSB
Sensitivity Change Due to Temperature			±0.01		%/°C
0 *g* OFFSET	Each axis				
0 *g* Output for X_{OUT}, Y_{OUT}		−150	0	+150	m*g*
0 *g* Output for Z_{OUT}		−250	0	+250	m*g*
0 *g* Output Deviation from Ideal, X_{OUT}, Y_{OUT}			±35		m*g*
0 *g* Output Deviation from Ideal, Z_{OUT}			±40		m*g*
0 *g* Offset vs. Temperature for X-, Y-Axes			±0.4		m*g*/°C
0 *g* Offset vs. Temperature for Z-Axis			±1.2		m*g*/°C
NOISE					
X-, Y-Axes	ODR = 100 Hz for ±2 *g*, 10-bit resolution or all *g*-ranges, full resolution		0.75		LSB rms
Z-Axis	ODR = 100 Hz for ±2 *g*, 10-bit resolution or all *g*-ranges, full resolution		1.1		LSB rms
OUTPUT DATA RATE AND BANDWIDTH	User selectable				
Output Data Rate (ODR)[3, 4, 5]		0.1		3200	Hz
SELF-TEST[6]					
Output Change in X-Axis		0.20		2.10	*g*
Output Change in Y-Axis		−2.10		−0.20	*g*
Output Change in Z-Axis		0.30		3.40	*g*
POWER SUPPLY					
Operating Voltage Range (V_S)		2.0	2.5	3.6	V
Interface Voltage Range ($V_{DD\ I/O}$)		1.7	1.8	V_S	V
Supply Current	ODR ≥ 100 Hz		140		µA
	ODR < 10 Hz		30		µA
Standby Mode Leakage Current			0.1		µA
Turn-On and Wake-Up Time[7]	ODR = 3200 Hz		1.4		ms

Parameter	Test Conditions	Min	Typ[1]	Max	Unit
TEMPERATURE					
Operating Temperature Range		−40		+85	°C
WEIGHT					
Device Weight			30		m*g*

[1] The typical specifications shown are for at least 68% of the population of parts and are based on the worst case of mean ±1 σ, except for 0 *g* output and sensitivity, which represents the target value. For 0 *g* offset and sensitivity, the deviation from the ideal describes the worst case of mean ±1 σ.
[2] Cross-axis sensitivity is defined as coupling between any two axes.
[3] Bandwidth is the −3 dB frequency and is half the output data rate, bandwidth = ODR/2.
[4] The output format for the 3200 Hz and 1600 Hz ODRs is different than the output format for the remaining ODRs. This difference is described in the Data Formatting of Upper Data Rates section.
[5] Output data rates below 6.25 Hz exhibit additional offset shift with increased temperature, depending on selected output data rate. Refer to the Offset Performance at Lowest Data Rates section for details.
[6] Self-test change is defined as the output (*g*) when the SELF_TEST bit = 1 (in the DATA_FORMAT register, Address 0x31) minus the output (*g*) when the SELF_TEST bit = 0. Due to device filtering, the output reaches its final value after 4 × τ when enabling or disabling self-test, where τ = 1/(data rate). The part must be in normal power operation (LOW_POWER bit = 0 in the BW_RATE register, Address 0x2C) for self-test to operate correctly.
[7] Turn-on and wake-up times are determined by the user-defined bandwidth. At a 100 Hz data rate, the turn-on and wake-up times are each approximately 11.1 ms. For other data rates, the turn-on and wake-up times are each approximately τ + 1.1 in milliseconds, where τ = 1/(data rate).

ABSOLUTE MAXIMUM RATINGS

Table 2.

Parameter	Rating
Acceleration	
Any Axis, Unpowered	10,000 *g*
Any Axis, Powered	10,000 *g*
V_S	−0.3 V to +3.9 V
$V_{DD\ I/O}$	−0.3 V to +3.9 V
Digital Pins	−0.3 V to $V_{DD\ I/O}$ + 0.3 V or 3.9 V, whichever is less
All Other Pins	−0.3 V to +3.9 V
Output Short-Circuit Duration (Any Pin to Ground)	Indefinite
Temperature Range	
Powered	−40°C to +105°C
Storage	−40°C to +105°C

Stresses at or above those listed under Absolute Maximum Ratings may cause permanent damage to the product. This is a stress rating only; functional operation of the product at these or any other conditions above those indicated in the operational section of this specification is not implied. Operation beyond the maximum operating conditions for extended periods may affect product reliability.

THERMAL RESISTANCE

Table 3. Package Characteristics

Package Type	θ_{JA}	θ_{JC}	Device Weight
14-Terminal LGA	150°C/W	85°C/W	30 mg

PACKAGE INFORMATION

The information in Figure 2 and Table 4 provide details about the package branding for the ADXL345. For a complete listing of product availability, see the Ordering Guide section.

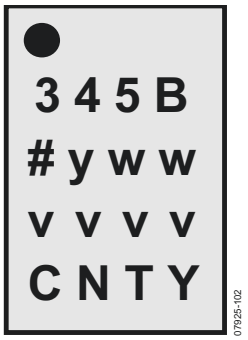


Figure 2. Product Information on Package (Top View)

Table 4. Package Branding Information

Branding Key	Field Description
345B	Part identifier for ADXL345
#	RoHS-compliant designation
yww	Date code
vvvv	Factory lot code
CNTY	Country of origin

ESD CAUTION

ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

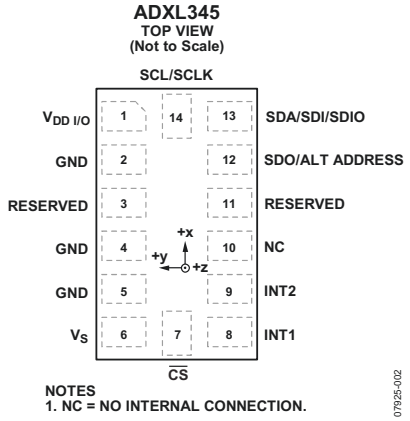


Figure 3. Pin Configuration (Top View)

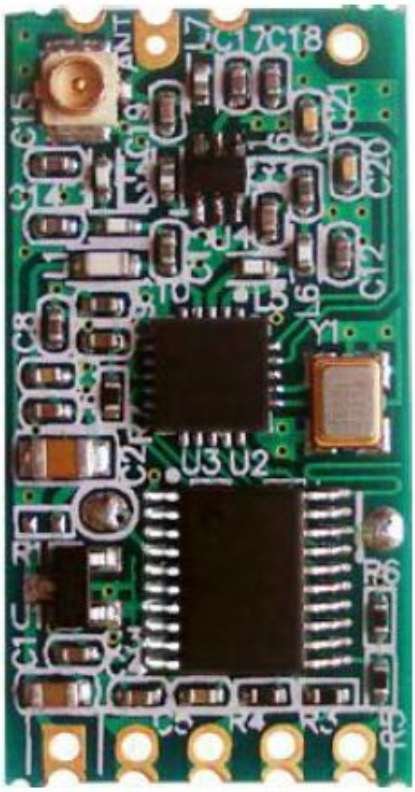
Table 5. Pin Function Descriptions

Pin No.	Mnemonic	Description
1	$V_{DD\ I/O}$	Digital Interface Supply Voltage.
2	GND	This pin must be connected to ground.
3	RESERVED	Reserved. This pin must be connected to V_S or left open.
4	GND	This pin must be connected to ground.
5	GND	This pin must be connected to ground.
6	V_S	Supply Voltage.
7	$\overline{CS}$	Chip Select.
8	INT1	Interrupt 1 Output.
9	INT2	Interrupt 2 Output.
10	NC	Not Internally Connected.
11	RESERVED	Reserved. This pin must be connected to ground or left open.
12	SDO/ALT ADDRESS	Serial Data Output (SPI 4-Wire)/Alternate I^2C Address Select (I^2C).
13	SDA/SDI/SDIO	Serial Data (I^2C)/Serial Data Input (SPI 4-Wire)/Serial Data Input and Output (SPI 3-Wire).
14	SCL/SCLK	Serial Communications Clock. SCL is the clock for I^2C, and SCLK is the clock for SPI.

ANEXO III. DATASHEET HC-12

HC-12 Wireless Serial Port Communication Module

User Manual V1.18



Product Application

- Wireless sensor
- Community building security
- Robot wireless control
- Industrial remote control and telemetering
- Automatic data acquisition
- Container information management
- POS system
- Wireless acquisition of gas meter data
- Vehicle keyless entry system
- PC wireless networking

……

Document: 2012/10

Product Features

- Long-distance wireless transmission (1,000m in open space/baud rate 5,000bps in the air)
- Working frequency range (433.4-473.0MHz, up to 100 communication channels)
- Maximum 100mW (20dBm) transmitting power (8 gears of power can be set)
- Three working modes, adapting to different application situations
- Built-in MCU, performing communication with external device through serial port
- The number of bytes transmitted unlimited to one time
- Update software version through serial port

Product Introduction

HC-12 wireless serial port communication module is a new-generation multichannel embedded wireless data transmission module. Its wireless working frequency band is 433.4-473.0MHz, multiple channels can be set, with the stepping of 400 KHz, and there are totally 100 channels. The maximum transmitting power of module is 100mW (20dBm), the receiving sensitivity is -117dBm at baud rate of 5,000bps in the air, and the communication distance is 1,000m in open space.

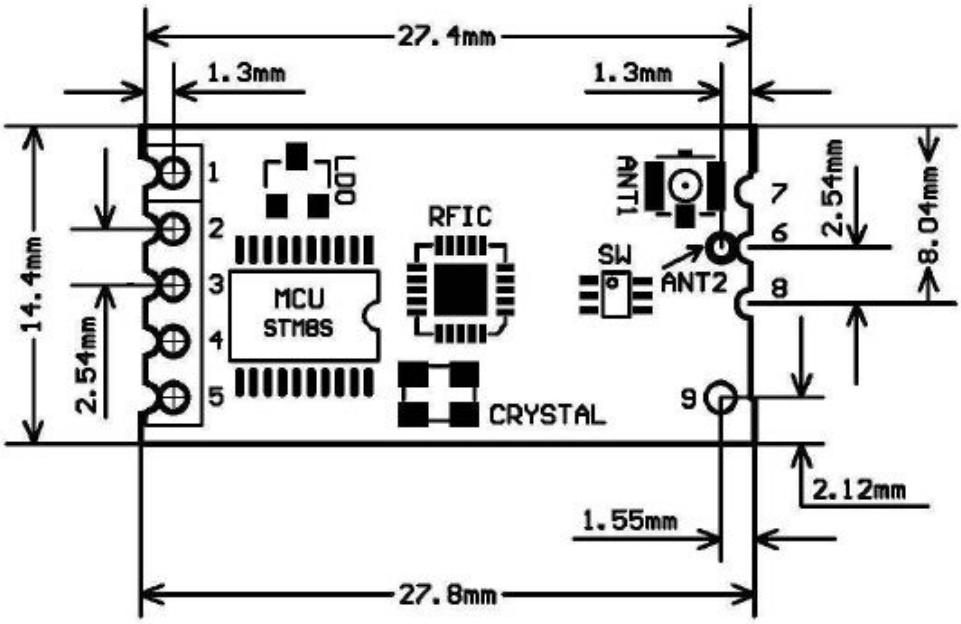
The module is encapsulated with stamp hole, can adopt patch welding, and its dimension is 27.8mm × 14.4mm × 4mm (including antenna cap, excluding spring antenna), so it is very convenient for customers to go into application system. There is a PCB antenna pedestal ANT1 on the module, and user can use external antenna of 433M frequency band through coaxial cable; there is also an antenna solder eye ANT2 in the module, and it is convenient for user to weld spring antenna. User could select one of these antennas according to use requirements.

There is MCU inside the module, and user don't need to program the module separately, and all transparent transmission mode is only responsible for receiving and sending serial port data, so it is convenient to use. The module adopts multiple serial port transparent transmission modes, and user could select them by AT command according to use requirements. The average working current of three modes FU1, FU2 and FU3 in idle state is 80μa, 3.6mA an 16mA respectively, and the maximum working current is 100mA (in transmitting state).

Product Configuration

Standard configuration of HC-12 module only contains one 433MHz-frequency-band wireless communication module with IPEX20279-001E-03 standard RF socket. The optional accessories are 433MHz-frequency-band spring antenna, IPEX-to-BNC coaxial cable and matching 433MHz-frequency-band omni-directional rubber antenna of BNC connector. User could purchase them according to use requirements.

Product Dimension



Definition of Pins

HC-12 module can adopt patch welding, or weld 2.54mm-spacing pin header, and directly insert it onto user's PCB. The module totally has nine pins and one RF antenna pedestal ANT1, and their definitions are as shown in the table below:

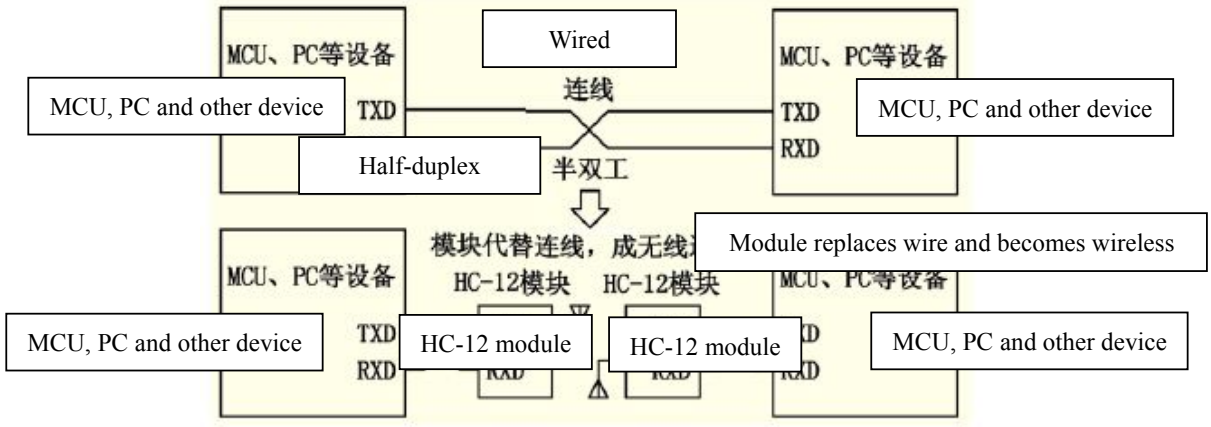
Pin	Definition	I/O direction	Note
1	VCC		Power supply input, DC3.2V-5.5V, with load capacity not less than 200mA. (Note: If the module is working in transmitting state for a long time, it is suggested that one 1N4007 diode should be connected in series when the power voltage is greater than 4.5V, to avoid heating of built-in LDO of module.)
2	GND		Common ground

3	RXD	Input, weak pull-up	URAT input port, TTL level; 1k resistance has been connected in series inside
4	TXD	Output	URAT output port, TTL level; 1k resistance has been connected in series inside
5	SET	Input, internal 10k pull-up resistance	Parameter setting control pin, valid for low level; 1k resistance has been connected in series inside
6	ANT	Input/output	433MHz antenna pin
7	GND		Common ground
8	GND		Common ground
9	NC		No connection, used in fixing, compatible with HC-11 module pin position
ANT1	ANT	Input/output	IPEX20279-001E-03 antenna socket
ANT2	ANT	Input/output	433MHz spring antenna solder eye

Pints 1-6 have two bonding pads respectively, and the outer half-hole bonding pad is used in patch welding. When the inner bonding pad ANT2 of Pin 6 is used in patch welding, the spring antenna can be welded with hands. The inner round-hole bonding pads of Pins 1-5 are used to weld 2.54mm-spacing pin header, and can be directly inserted onto user's PCB socket.

Wireless serial port transparent transmission

(1) Simple introduction of working p principle



As shown in the above figure, HC-12 module is used in physical wiring when replacing half duplex communication. The left device sends serial port data to module, and after RXD port of left module receives the serial port data, it will automatically send the data into the air via radio wave. The right module can automatically receive the data, and restore, from TXD, the serial port data originally sent by the left device. It is the same from right to left. Only half duplex state is available between modules, and they cannot receive and send data at the same time.

(2) Serial port transparent transmission

HC-12 module has three serial port transparent transmission modes, expressed with FU1, FU2 and FU3 respectively. In the use, all modes are only responsible for receiving and sending serial port data rather than wireless transmission. The default working mode of system is in FU3 full-speed mode, and in this mode, the baud rate in this air can be automatically adjusted according to baud rate of serial port, and the communication distance will be the farthest at the low baud rate. Different modes cannot transmit data to each other, and user could select the optimal mode according to practical circumstances.

The modules are usually used in pairs, and transmit data by means of half duplex. Meanwhile, the transparent transmission mode, serial port baud rate, and wireless communication channel of two paired modules shall be set to be the same. The default setting is FU3, 9,600bps (8-dibit data, no check, one stop bit), CH001 (433.4MHz).

Use the number of bytes continuously sent to serial port of module unlimited to one time. However, considering ambient interference and other factors, if thousands of data size is sent continuously at a time, some number of bytes may be lost. Therefore, the upper computer shall have response and resending mechanism, to avoid information loss.

(3) Three serial port transparent transmission modes

When HC-12 module leaves the factory, its default serial port transparent transmission mode is FU3. Then the module is in full-speed state, and the idle current is about

16mA. In this mode, the module can automatically adjust the baud rate of wireless transmission in the air according to serial port baud rate, and the corresponding relationship is as shown in the table below:

Serial port baud rate	1,200bps	2,400bps	4,800bps	9,600bps	19,200bps	38,400bps	57,600bps	115,200bps
Baud rate in the air	5,000bps		15,000bps		58,000bps		236,000bps	

To get the farthest communication distance, the serial port baud rate can be set to be low. For short-time transmission of mass data, set the serial port baud rate to be high, but the communication distance will be reduced accordingly.

The receiving sensitivity of module at different baud rates in the air is as shown in the table below:

Baud rate in the air	5,000bps	15,000bps	58,000bps	236,000bps
Wireless receiving sensitivity	-117dBm	-112dBm	-107dBm	-100dBm

Generally, every time the receiving sensitivity is reduced by 6dB, the communication distance will be reduced by half.

When “SET” pin of module is at low level, the serial port transparent transmission mode can be set through AT command (see the introduction in the following chapter for details).

FU1 mode is relatively power saving mode, and the idle working current of this mode is about 3.6mA. In this mode, the module can also set eight types of serial port baud rate as shown in the above table, but the baud rate in the air is uniform, 250,000bps.
FU2 mode is power saving mode, and the idle working current of this mode is about 80μA. In this mode, the module only supports baud rates of 1,200bps, 2,400bps and 4,800bps, and the baud rate in the air is uniform, 250,000bps. If the module is set to be other serial port baud rate, the module cannot conduct communication normally.

Meanwhile, when the module is set to be FU2 mode in FU1 and FU3 mode, the baud rate exceeding 4,800bps will be automatically reduced to be 4,800bps. In FU2 mode, the sending time interval of data package cannot be too short; otherwise, the data will be lost. It is suggested that the sending time interval of data package should not be less than 1sec.

The following gives some characteristics reference values of various modes:

Mode	FU1	FU2	FU3	Remark
Idle current	3.6mA	80μA	16mA	Average value
Transmission time delay	15-25mS	500mS	4-80mS	Sending one byte
Loopback test time delay 1	31mS			Serial port baud rate 9,600, sending one byte
Loopback test time delay 2	31mS			Serial port baud rate 9,600, sending ten bytes

Note: Loopback test time delay means the duration from the time of, after conducting short circuit on TX and RX pins of one module and sending serial port data to the other module, starting to send serial port data to the other module to the time that the returned data appear at TX pin of the other module.

Module Parameter Setting AT Command

AT command is used to set the module parameters and switch the module functions, and after setting, it will be valid only after exiting from setting state. Meanwhile, modification of parameters and functions will not be lost in case of power failure.

(1) Command mode entering

The first way to enter: in normal use (energized), put Pin 5 “SET” in low level;
The second way to enter: disconnect power supply, first put Pin 5 “Set” in low level, and then energize it;

Either of the above two ways can make the module enter AT command mode; release it (not put pin “SET” in low level), and exit from the command mode. If the module function is changed after exiting from command mode, it will be switched to corresponding functional status.

In the second way, the module enters AT in the serial port format of 9,600, N, 1 constantly.

(2) Command instruction

①. AT

Test command.

e.g.:
Send “AT” command to module, and the module returns “OK”.

②AT+Bxxxx
Change the serial port baud rate. The baud rate can be set to be 1,200bps, 2,400bps, 4,800bps, 9,600bps, 19,200bps, 38,400bps, 57,600bps, and 115,200bps. The default value is 9,600bps.
e.g.: To set serial port baud rate of module to be 19,200bps, first send “AT+B19200” command to module, and the module returns “OK+B19200”.

③AT+Cxxxx
Change wireless communication channel, optional from 001 to 127 (for the wireless channel exceeding 100, the communication distance cannot be ensured). The default value of wireless channel is 001, and the working frequency is 433.4MHz. The channel stepping is 400KHz, and the working frequency of Channel 100 is 473.0MHz.
e.g.:
To set the module to work at Channel 21, first send “AT+C021” command to the module, and the module returns “COK+C021”. After exiting from the command mode, the module will work at Channel 21, and the working frequency is 441.4MHz.
Note: As the wireless receiving sensitivity of HC-12 module is relatively high, when the serial port baud rate is greater than 9,600bps, five adjacent channels shall be staggered to use. When the serial port baud rate is not greater than 9,600bps, in short-distance (within 10m) communication, also five adjacent channels shall be staggered to use.

④AT+FUx
Change serial port transparent transmission mode of module and three modes are available, namely, FU1, FU2 and FU3. The default mode of module is FU3, and only when serial port transparent transmission mode of two modules is set to be the same, can normal communication be available. For detailed introduction, please see the above “wireless serial port transparent transmission”.
e.g.:
Send “AT+FU1” to module, and the module returns “AT+0K”.

⑤AT+Px
Set transmitting power of module, x is optional from 1 to 8, and the corresponding transmitting power of module is as shown below:

x value	1	2	3	4	5	6	7	8
Transmitting power of module (dBm)	-1	2	5	8	11	14	17	20

The default value is 8, and the higher the transmitting power is, the farther the communication distance is. When the transmitting power level is set to be 1, the transmitting power is the minimum. Generally speaking, every time the transmitting power is reduced by 6dB, the communication distance will be reduced by half.
e.g.:
Send “AT+P5” command to module, and the module returns “OK+P5”. After exiting from the command code, the transmitting power of module is +11dBm.

⑥AT+Ry
Obtain single parameter of module, y is any letter among B, C, F and P, respectively representing: baud rate, communication channel, serial port transparent transmission mode, and transmitting power.
Example 1:
Send “AT+RB” to module, and if the module returns “OK+B9600”, it is inquired that the serial port baud rate of module is 9,600bps.
Example 2:
Send “AT+RC” command to module, and if the module returns “OK+RC001”, it is inquired that the communication channel of module is 001.
Example 3:
Send “AT+RF” command to module, and if the module returns “OK+FU3”, it is inquired that the module is working in serial port transparent transmission mode 3.
Example 4:
Send “AT+RP” command to module, and if the module returns “OK+RP: +20dBm”, it is inquired that the transmitting power of module is +20dBm.

⑦AT+RX
Obtain all parameters of module. Return serial port transparent transmission mode, serial port baud rate, communication channel, and transmitting power in order.
e.g.:
Send “AT+RX” command to module, and the module returns “OK+FU3\r\n OK+B9600\r\n OK+C001\r\n OK+RP: +20dBm\r\n”. (“\r\n” means return\newline)

⑧AT+Uxxx
Set data bits, check bit and stop bit of serial port communication. For check bit, N means no check, O means odd check, and E means even check. For stop bit, 1 means one stop bit, 2 means two stop bits, and 3 means 1.5 stop bits.
e.g.:
To send serial port format to be eight data bits, odd check, and one stop bit, please Send “AT+U8O1” to module, and the module returns “OK+U8O1”.

⑨AT+V
Inquire firmware version information of module.
e.g.:
Send “AT+V” command to module, and the module returns “HC-12_V1.1”.

⑩AT+SLEEP

After receiving the command, the module enters sleep mode after exiting from AT, the working current is about 22μA, and this mode doesn't allow serial port data transmission. Then enter AT setting state again, and the module will exit from sleep mode automatically.

e.g.:

When wireless data transmission is not needed, to save power, send "AT+SLEEP" command to module, and the module returns "OK+SLEEP".

⑾AT+DEFAULT

Set serial port baud rate, communication channel, and serial port transparent transmission mode to be default value.

e.g.:

Send "AT+DEFAULT" to module, and the module returns "OK+DEFAULT", and the default vale is restored. The serial port baud rate is 9,600bps, communication channel is C001, and serial port transparent transmission mode is FU3.

⑿AT+UPDATE

Put the module in the status of waiting for software update.

After sending the command, the module will not respond to command any more, until it is re-energized.

After sending the command, please close the serial port assistant, and turn on HC-1X updater to update the software. For detailed operating method, please refer to the following "software update" introduction.

ANEXO IV. DATASHEET NEO-6

GPS

locate, communicate, accelerate

NEO-6

u-blox 6 GPS Modules

Data Sheet

Abstract

Technical data sheet describing the cost effective, high-performance u-blox 6 based NEO-6 series of GPS modules, that brings the high performance of the u-blox 6 positioning engine to the miniature NEO form factor.

These receivers combine a high level of integration capability with flexible connectivity options in a small package. This makes them perfectly suited for mass-market end products with strict size and cost requirements.

16.0 x 12.2 x 2.4 mm

www.u-blox.com

Document Information	
Title	NEO-6
Subtitle	u-blox 6 GPS Modules
Document type	Data Sheet
Document number	GPS.G6-HW-09005-E
Document status	

Document status information	
Objective Specification	This document contains target values. Revised and supplementary data will be published later.
Advance Information	This document contains data based on early testing. Revised and supplementary data will be published later.
Preliminary	This document contains data from product verification. Revised and supplementary data may be published later.
Released	This document contains the final product specification.

This document applies to the following products:

Name	Type number	ROM/FLASH version	PCN reference
NEO-6G	NEO-6G-0-001	ROM7.03	UBX-TN-11047-1
NEO-6Q	NEO-6Q-0-001	ROM7.03	UBX-TN-11047-1
NEO-6M	NEO-6M-0-001	ROM7.03	UBX-TN-11047-1
NEO-6P	NEO-6P-0-000	ROM6.02	N/A
NEO-6V	NEO-6V-0-000	ROM7.03	N/A
NEO-6T	NEO-6T-0-000	ROM7.03	N/A

Contents

1 Functional description

1.1 Overview

The NEO-6 module series is a family of stand-alone GPS receivers featuring the high performance u-blox 6 positioning engine. These flexible and cost effective receivers offer numerous connectivity options in a miniature 16 x 12.2 x 2.4 mm package. Their compact architecture and power and memory options make NEO-6 modules ideal for battery operated mobile devices with very strict cost and space constraints.

The 50-channel u-blox 6 positioning engine boasts a Time-To-First-Fix (TTFF) of under 1 second. The dedicated acquisition engine, with 2 million correlators, is capable of massive parallel time/frequency space searches, enabling it to find satellites instantly. Innovative design and technology suppresses jamming sources and mitigates multipath effects, giving NEO-6 GPS receivers excellent navigation performance even in the most challenging environments.

1.2 Product features

Model	Type					Supply		Interfaces				Features						
	GPS	PPP	Timing	Raw Data	Dead Reckoning	1.75 V - 2.0 V	2.7 V - 3.6 V	UART	USB	SPI	DDC (I²C compliant)	Programmable (Flash) FW update	TCXO	RTC crystal	Antenna supply and supervisor	Configuration pins	Timepulse	External interrupt/ Wakeup
NEO-6G	●					●		●	●	●	●		●	●	o	3	1	●
NEO-6Q	●						●	●	●	●	●		●	●	o	3	1	●
NEO-6M	●						●	●	●	●	●			●	o	3	1	●
NEO-6P	●	●		●			●	●	●	●	●			●	o	3	1	●
NEO-6V	●				●		●	●	●	●	●			●	o	3	1	●
NEO-6T	●		●	●			●	●	●	●	●		●	●	o	3	1	●

o = Requires external components and integration on application processor

Table 1: Features of the NEO-6 Series

☞ All NEO-6 modules are based on GPS chips qualified according to AEC-Q100. See Chapter 5.1 for further information.

1.3 GPS performance

Parameter	Specification			
Receiver type	50 Channels GPS L1 frequency, C/A Code SBAS: WAAS, EGNOS, MSAS			
Time-To-First-Fix[1]		NEO-6G/Q/T	NEO-6M/V	NEO-6P
	Cold Start[2]	26 s	27 s	32 s
	Warm Start[2]	26 s	27 s	32 s
	Hot Start[2]	1 s	1 s	1 s
	Aided Starts[3]	1 s	<3 s	<3 s
Sensitivity[4]		NEO-6G/Q/T	NEO-6M/V	NEO-6P
	Tracking & Navigation	-162 dBm	-161 dBm	-160 dBm
	Reacquisition[5]	-160 dBm	-160 dBm	-160 dBm
	Cold Start (without aiding)	-148 dBm	-147 dBm	-146 dBm
	Hot Start	-157 dBm	-156 dBm	-155 dBm
Maximum Navigation update rate		NEO-6G/Q/M/T	NEO-6P/V	
		5Hz	1 Hz	
Horizontal position accuracy[6]	GPS	2.5 m		
	SBAS	2.0 m		
	SBAS + PPP[7]	< 1 m (2D, R50)[8)]		
	SBAS + PPP[7]	< 2 m (3D, R50)[8]		
Configurable Timepulse frequency range		NEO-6G/Q/M/P/V	NEO-6T	
		0.25 Hz to 1 kHz	0.25 Hz to 10 MHz	
Accuracy for Timepulse signal	RMS	30 ns		
	99%	<60 ns		
	Granularity	21 ns		
	Compensated[9]	15 ns		
Velocity accuracy[6]		0.1m/s		
Heading accuracy[6]		0.5 degrees		
Operational Limits	Dynamics	≤ 4 g		
	Altitude[10]	50,000 m		
	Velocity[10]	500 m/s		

Table 2: NEO-6 GPS performance

[1] All satellites at -130 dBm
[2] Without aiding
[3] Dependent on aiding data connection speed and latency
[4] Demonstrated with a good active antenna
[5] For an outage duration ≤10s
[6] CEP, 50%, 24 hours static, -130dBm, SEP: <3.5m
[7] NEO-6P only
[8] Demonstrated under following conditions: 24 hours, stationary, first 600 seconds of data discarded. HDOP < 1.5 during measurement period, strong signals. Continuous availability of valid SBAS correction data during full test period.
[9] Quantization error information can be used with NEO-6T to compensate the granularity related error of the timepulse signal
[10] Assuming Airborne <4g platform

1.4 Block diagram

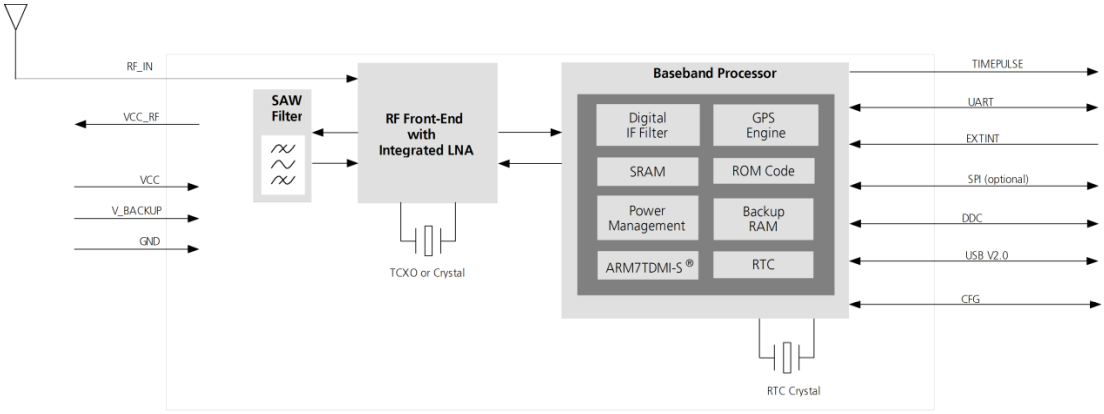


Figure 1: Block diagram (For available options refer to the product features table in section 1.2).

1.5 Assisted GPS (A-GPS)

Supply of aiding information like ephemeris, almanac, rough last position and time and satellite status and an optional time synchronization signal will reduce time to first fix significantly and improve the acquisition sensitivity. All NEO-6 modules support the u-blox AssistNow Online and AssistNow Offline A-GPS services[11] and are OMA SUPL compliant.

1.6 AssistNow Autonomous

AssistNow Autonomous provides functionality similar to Assisted-GPS without the need for a host or external network connection. Based on previously broadcast satellite ephemeris data downloaded to and stored by the GPS receiver, AssistNow Autonomous automatically generates accurate satellite orbital data ("AssistNow Autonomous data") that is usable for future GPS position fixes. AssistNow Autonomous data is reliable for up to 3 days after initial capture.

u-blox' AssistNow Autonomous benefits are:

- Faster position fix
- No connectivity required
- Complementary with AssistNow Online and Offline services
- No integration effort, calculations are done in the background

☞ For more details see the u-blox 6 Receiver Description including Protocol Specification [2].

[11] AssistNow Offline requires external memory.

1.7 Precision Timing

1.7.1 Time mode

NEO-6T provides a special Time Mode to provide higher timing accuracy. The NEO-6T is designed for use with stationary antenna setups. The Time Mode features three different settings described in Table 3: Disabled, Survey-In and Fixed Mode. For optimal performance entering the position of the antenna (when known) is recommended as potential source of errors will be reduced.

Time Mode Settings	Description
Disabled	Standard PVT operation
Survey-In	The GPS receiver computes the average position over an extended time period until a predefined maximum standard deviation has been reached. Afterwards the receiver will be automatically set to Fixed Mode and the timing features will be activated.
Fixed Mode	In this mode, a fixed 3D position and known standard deviation is assumed and the timing features are activated. Fixed Mode can either be activated directly by feeding pre-defined position coordinates (ECEF - Earth Center Earth Fixed format) or by performing a Survey-In. In Fixed mode, the timing errors in the TIMEPULSE signal which otherwise result from positioning errors are eliminated. Single-satellite operation is supported. For details, please refer to the u-blox 6 Receiver Description including Protocol Specification [2].

Table 3: Time mode settings

1.7.2 Timepulse and frequency reference

NEO-6T comes with a timepulse output which can be configured from 0.25 Hz up to 10 MHz. The timepulse can either be used for time synchronization (i.e. 1 pulse per second) or as a reference frequency in the MHz range. A timepulse in the MHz range provides excellent long-term frequency accuracy and stability.

1.7.3 Time mark

NEO-6T can be used for precise time measurements with sub-microsecond resolution using the external interrupt (EXTINT0). Rising and falling edges of these signals are time-stamped to the GPS or UTC time and counted. The Time Mark functionality can be enabled with the UBX-CFG-TM2 message

For details, please refer to the u-blox 6 Receiver Description including Protocol Specification [2].

1.8 Raw data

Raw data output is supported at an update rate of 5 Hz on the NEO-6T and NEO-6P. The UBX-RXM-RAW message includes carrier phase with half-cycle ambiguity resolved, code phase and Doppler measurements, which can be used in external applications that offer precision positioning, real-time kinematics (RTK) and attitude sensing.

1.9 Automotive Dead Reckoning

Automotive Dead Reckoning (ADR) is u-blox' industry proven off-the-shelf Dead Reckoning solution for tier-one automotive customers. u-blox' ADR solution combines GPS and sensor digital data using a tightly coupled Kalman filter. This improves position accuracy during periods of no or degraded GPS signal.

The NEO-6V provides ADR functionality over its software sensor interface. A variety of sensors (such as wheel ticks and gyroscope) are supported, with the sensor data received via UBX messages from the application processor. This allows for easy integration and a simple hardware interface, lowering costs. By using digital sensor data available on the vehicle bus, hardware costs are minimized since no extra sensors are required for Dead Reckoning functionality. ADR is designed for simple integration and easy configuration of different sensor options (e.g. with or without gyroscope) and vehicle variants, and is completely self-calibrating.

For more details contact the u-blox support representative nearest you to receive dedicated u-blox 6 Receiver Description Including Protocol Specification [3].

1.10 Precise Point Positioning

u-blox' industry proven PPP algorithm provides extremely high levels of position accuracy in static and slow moving applications, and makes the NEO-6P an ideal solution for a variety of high precision applications such as surveying, mapping, marine, agriculture or leisure activities.

Ionospheric corrections such as those received from local SBAS[12] geostationary satellites (WAAS, EGNOS, MSAS) or from GPS enable the highest positioning accuracy with the PPP algorithm. The maximum improvement of positioning accuracy is reached with PPP+SBAS and can only be expected in an environment with unobstructed sky view during a period in the order of minutes.

1.11 Oscillators

NEO-6 GPS modules are available in Crystal and TCXO versions. The TCXO allows accelerated weak signal acquisition, enabling faster start and reacquisition times.

1.12 Protocols and interfaces

Protocol	Type
NMEA	Input/output, ASCII, 0183, 2.3 (compatible to 3.0)
UBX	Input/output, binary, u-blox proprietary
RTCM	Input, 2.3

Table 4: Available protocols

All listed protocols are available on UART, USB, and DDC. For specification of the various protocols see the u-blox 6 Receiver Description including Protocol Specification [2].

1.12.1 UART

NEO-6 modules include one configurable UART interface for serial communication (for information about configuration see section 1.15).

1.12.2 USB

NEO-6 modules provide a USB version 2.0 FS (Full Speed, 12Mbit/s) interface as an alternative to the UART. The pull-up resistor on USB_DP is integrated to signal a full-speed device to the host. The VDDUSB pin supplies the USB interface. u-blox provides a Microsoft® certified USB driver for Windows XP, Windows Vista and Windows 7 operating systems.

1.12.3 Serial Peripheral Interface (SPI)

The SPI interface allows for the connection of external devices with a serial interface, e.g. serial flash to save configuration and AssistNow Offline A-GPS data or to interface to a host CPU. The interface can be operated in master or slave mode. In master mode, one chip select signal is available to select external slaves. In slave mode a single chip select signal enables communication with the host.

☞ The maximum bandwidth is 100kbit/s.

[12] Satellite Based Augmentation System

1.12.4 Display Data Channel (DDC)

The I²C compatible DDC interface can be used either to access external devices with a serial interface EEPROM or to interface with a host CPU. It is capable of master and slave operation. The DDC interface is I²C Standard Mode compliant. For timing parameters consult the I²C standard.

☞ The DDC Interface supports serial communication with u-blox wireless modules. See the specification of the applicable wireless module to confirm compatibility.

☞ The maximum bandwidth is 100kbit/s.

1.12.4.1 External serial EEPROM

NEO-6 modules allow an optional external serial EEPROM to be connected to the DDC interface. This can be used to store Configurations permanently.

☞ For more information see the LEA-6/NEO-6/MAX-6 Hardware Integration Manual [1].

⚠ **Use caution when implementing since forward compatibility is not guaranteed.**

1.13 Antenna

NEO-6 modules are designed for use with passive and active[13] antennas.

Parameter	Specification	
Antenna Type		Passive and active antenna
Active Antenna Recommendations	Minimum gain	15 dB (to compensate signal loss in RF cable)
	Maximum gain	50 dB
	Maximum noise figure	1.5 dB

Table 5: Antenna Specifications for all NEO-6 modules

1.14 Power Management

u-blox receivers support different power modes. These modes represent strategies of how to control the acquisition and tracking engines in order to achieve either the best possible performance or good performance with reduced power consumption.

☞ For more information about power management strategies, see the u-blox 6 Receiver Description including Protocol Specification [2].

1.14.1 Maximum Performance Mode

During a Cold start, a receiver in Maximum Performance Mode continuously deploys the acquisition engine to search for all satellites. Once the receiver has a position fix (or if pre-positioning information is available), the acquisition engine continues to be used to search for all visible satellites that are not being tracked.

1.14.2 Eco Mode

During a Cold start, a receiver in Eco Mode works exactly as in Maximum Performance Mode. Once a position can be calculated and a sufficient number of satellites are being tracked, the acquisition engine is powered off resulting in significant power savings. The tracking engine continuously tracks acquired satellites and acquires other available or emerging satellites.

☞ Note that even if the acquisition engine is powered off, satellites continue to be acquired.

[13] For information on using active antennas with NEO-6 modules, see the LEA-6/NEO-6 Hardware Integration Manual [1].

1.14.3 Power Save Mode

Power Save Mode (PSM) allows a reduction in system power consumption by selectively switching parts of the receiver on and off.

Power Save mode is not available with NEO-6P, NEO-6T and NEO-6V.

1.15 Configuration

1.15.1 Boot-time configuration

NEO-6 modules provide configuration pins for boot-time configuration. These become effective immediately after start-up. Once the module has started, the configuration settings can be modified with UBX configuration messages. The modified settings remain effective until power-down or reset. If these settings have been stored in battery-backup RAM, then the modified configuration will be retained, as long as the backup battery supply is not interrupted.

NEO-6 modules include both **CFG_COM0** and **CFG_COM1** pins and can be configured as seen in Table 6. Default settings in bold.

CFG_COM1	CFG_COM0	Protocol	Messages	UARTBaud rate	USB power
1	**1**	**NMEA**	**GSV, RMC, GSA, GGA, GLL, VTG, TXT**	**9600**	**BUS Powered**
1	0	NMEA	GSV, RMC, GSA, GGA, GLL, VTG, TXT	38400	Self Powered
0	1	NMEA	GSV[14], RMC, GSA, GGA, VTG, TXT	4800	BUS Powered
0	0	UBX	NAV-SOL, NAV-STATUS, NAV-SVINFO, NAV-CLOCK, INF, MON-EXCEPT, AID-ALPSERV	57600	BUS Powered

Table 6: Supported COM settings

NEO-6 modules include a **CFG_GPS0** pin, which enables the boot-time configuration of the power mode. These settings are described in Table 7. Default settings in bold.

CFG_GPS0	Power Mode
0	Eco Mode
1	**Maximum Performance Mode**

Table 7: Supported CFG_GPS0 settings

Static activation of the **CFG_COM** and **CFG_GPS** pins is not compatible with use of the SPI interface.

1.16 Design-in

In order to obtain the necessary information to conduct a proper design-in, u-blox strongly recommends consulting the LEA-6/NEO-6/MAX-6 Hardware Integration Manual [1].

[14] Every 5th fix.

2 Pin Definition

2.1 Pin assignment

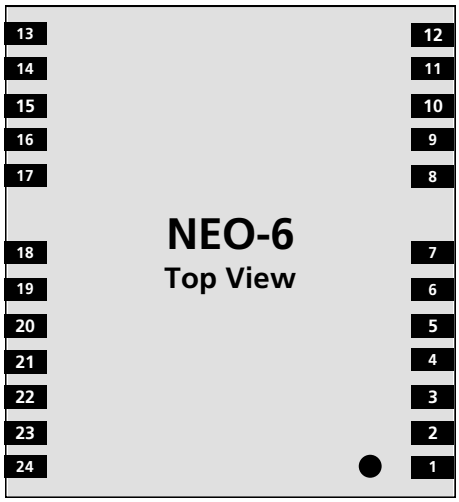


Figure 2 Pin Assignment

No	Module	Name	I/O	Description
1	All	Reserved	I	Reserved
2	All	SS_N	I	SPI Slave Select
3	All	TIMEPULSE	O	Timepulse (1PPS)
4	All	EXTINT0	I	External Interrupt Pin
5	All	USB_DM	I/O	USB Data
6	All	USB_DP	I/O	USB Data
7	All	VDDUSB	I	USB Supply
8	All	Reserved		See Hardware Integration Manual Pin 8 and 9 must be connected together.
9	All	VCC_RF	O	Output Voltage RF section Pin 8 and 9 must be connected together.
10	All	GND	I	Ground
11	All	RF_IN	I	GPS signal input
12	All	GND	I	Ground
13	All	GND	I	Ground
14	All	MOSI/CFG_COM0	O/I	SPI MOSI / Configuration Pin. Leave open if not used.
15	All	MISO/CFG_COM1	I	SPI MISO / Configuration Pin. Leave open if not used.
16	All	CFG_GPS0/SCK	I	Power Mode Configuration Pin / SPI Clock. Leave open if not used.
17	All	Reserved	I	Reserved
18	All	SDA2	I/O	DDC Data
19	All	SCL2	I/O	DDC Clock
20	All	TxD1	O	Serial Port 1
21	All	RxD1	I	Serial Port 1

No	Module	Name	I/O	Description
22	All	V_BCKP	I	Backup voltage supply
23	All	VCC	I	Supply voltage
24	All	GND	I	Ground

Table 8: Pinout

☞ Pins designated Reserved should not be used. For more information about Pinouts see the LEA-6/NEO-6/MAX-6 Hardware Integration Manual [1].

3 Electrical specifications

3.1 Absolute maximum ratings

Parameter	Symbol	Module	Min	Max	Units	Condition
Power supply voltage	VCC	NEO-6G	-0.5	2.0	V	
		NEO-6Q, 6M, 6P, 6V, 6T	-0.5	3.6	V	
Backup battery voltage	V_BCKP	All	-0.5	3.6	V	
USB supply voltage	VDDUSB	All	-0.5	3.6	V	
Input pin voltage	Vin	All	-0.5	3.6	V	
	Vin_usb	All	-0.5	VDDUSB	V	
DC current trough any digital I/O pin (except supplies)	Ipin			10	mA	
VCC_RF output current	ICC_RF	All		100	mA	
Input power at RF_IN	Prfin	NEO-6Q, 6M, 6G, 6V, 6T		15	dBm	source impedance = 50Ω, continuous wave
		NEO-6P		-5	dBm	
Storage temperature	Tstg	All	-40	85	°C	

Table 9: Absolute maximum ratings

GPS receivers are Electrostatic Sensitive Devices (ESD) and require special precautions when handling. For more information see chapter 6.4.

Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. The product is not protected against overvoltage or reversed voltages. If necessary, voltage spikes exceeding the power supply voltage specification, given in table above, must be limited to values within the specified boundaries by using appropriate protection diodes. For more information see the *LEA-6/NEO-6/MAX-6 Hardware Integration Manual* [1].

3.2 Operating conditions

☞ All specifications are at an ambient temperature of 25°C.

Parameter	Symbol	Module	Min	Typ	Max	Units	Condition
Power supply voltage	VCC	NEO-6G	1.75	1.8	1.95	V	
		NEO-6Q/M NEO-6P/V/T	2.7	3.0	3.6	V	
Supply voltage USB	VDDUSB	All	3.0	3.3	3.6	V	
Backup battery voltage	V_BCKP	All	1.4		3.6	V	
Backup battery current	I_BCKP	All		22		µA	V_BCKP = 1.8 V, VCC = 0V
Input pin voltage range	Vin	All	0		VCC	V	
Digital IO Pin Low level input voltage	Vil	All	0		0.2*VCC	V	
Digital IO Pin High level input voltage	Vih	All	0.7*VCC		VCC	V	
Digital IO Pin Low level output voltage	Vol	All			0.4	V	Iol=4mA
Digital IO Pin High level output voltage	Voh	All	VCC -0.4			V	Ioh=4mA
USB_DM, USB_DP	VinU	All	Compatible with USB with 22 Ohms series resistance				
VCC_RF voltage	VCC_RF	All		VCC-0.1		V	
VCC_RF output current	ICC_RF	All			50	mA	
Antenna gain	Gant	All			50	dB	
Receiver Chain Noise Figure	NFtot	All		3.0		dB	
Operating temperature	Topr	All	-40		85	°C	

Table 10: Operating conditions

☞ Operation beyond the specified operating conditions can affect device reliability.

3.3 Indicative power requirements

Table 11 lists examples of the total system supply current for a possible application.

Parameter	Symbol	Module	Min	Typ	Max	Units	Condition
Max. supply current [15]	Iccp	All			67	mA	VCC = 3.6 V[16] / 1.95 V[17]
Average supply current[18]	Icc Acquisition	All		47[19]		mA	VCC = 3.0 V[16] / 1.8 V[17]
	Icc Tracking (Max Performance mode)	NEO-6G/Q/T		40[20]		mA	
		NEO-6M/P/V		39[20]		mA	
	Icc Tracking (Eco mode)	NEO-6G/Q/T		38[20]		mA	
		NEO-6M/P/V		37[20]		mA	
	Icc Tracking (Power Save mode / 1 Hz)	NEO-6G/Q		12[20]		mA	
		NEO-6M		11[20]		mA	

Table 11: Indicative power requirements

☞ Values in Table 11 are provided for customer information only as an example of typical power requirements. Values are characterized on samples, actual power requirements can vary depending on FW version used, external circuitry, number of SVs tracked, signal strength, type of start as well as time, duration and conditions of test.

[15] Use this figure to dimension maximum current capability of power supply. Measurement of this parameter with 1 Hz bandwidth.
[16] NEO-6Q, NEO-6M, NEO-6P, NEO-6V, NEO-6T
[17] NEO-6G
[18] Use this figure to determine required battery capacity.
[19] >8 SVs in view, CNo >40 dBHz, current average of 30 sec after cold start.
[20] With strong signals, all orbits available. For Cold Starts typical 12 min after first fix. For Hot Starts typical 15 s after first fix.

3.4 SPI timing diagrams

In order to avoid a faulty usage of the SPI, the user needs to comply with certain timing conditions. The following signals need to be considered for timing constraints:

Symbol	Description
SS_N	Slave Select signal
SCK	Slave Clock signal

Table 12: Symbol description

Figure 3: SPI timing diagram

3.4.1 Timing recommendations

Parameter	Description	Recommendation
t_{INIT}	Initialization Time	500 µs
t_{DES}	Deselect Time	1 ms
Bitrate		100 kbit/s

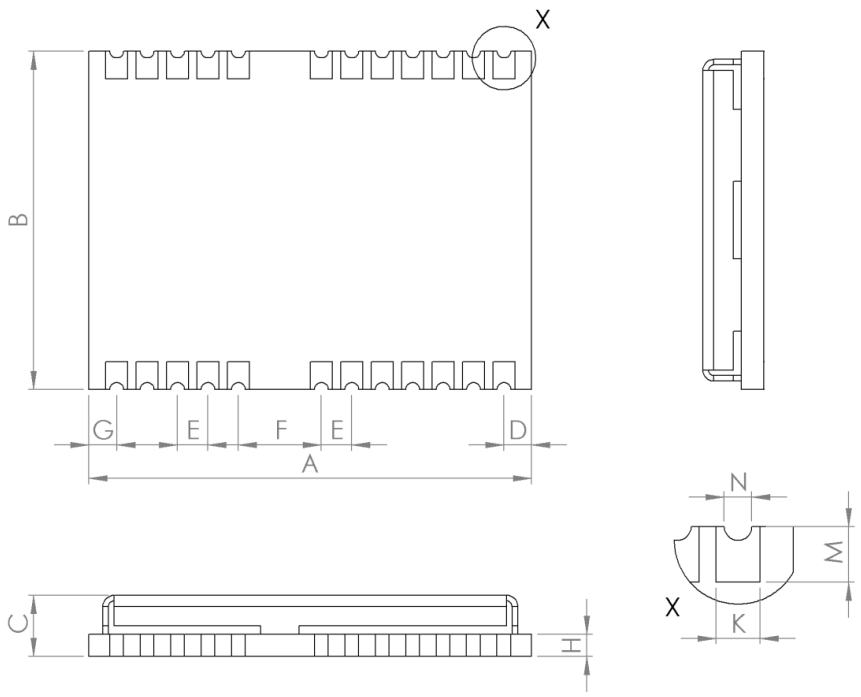
Table 13: SPI timing recommendations

The values in the above table result from the requirement of an error-free transmission. By allowing just a few errors, the byte rate could be increased considerably. These timings – and therefore the byte rate – could also be improved by disabling other interfaces, e.g. the UART.

The maximum bandwidth is 100 kbit/s[21].

[21] This is a theoretical maximum, the protocol overhead is not considered.

4 Mechanical specifications



Symbol	Min. (mm)	Typ. (mm)	Max. (mm)
A	15.9	16.0	16.6
B	12.1	12.2	12.3
C	2.2	2.4	2.6
D	0.9	1.0	1.3
E	1.0	1.1	1.2
F	2.9	3.0	3.1
G	0.9	1.0	1.3
H		0.82	
K	0.7	0.8	0.9
M	0.8	0.9	1.0
N	0.4	0.5	0.6
Weight		1.6g	

Figure 4: Dimensions

For information regarding the Paste Mask and Footprint see the LEA-6/NEO-6/MAX-6 Hardware Integration Manual [1].

ANEXO V. DATASHEET HT1621

HT1621

RAM Mapping 32×4 LCD Controller for I/O MCU

Technical Document

- Tools Information
- FAQs
- Application Note

Features

- Operating voltage: 2.4V~5.2V
- Built-in 256kHz RC oscillator
- External 32.768kHz crystal or 256kHz frequency source input
- Selection of 1/2 or 1/3 bias, and selection of 1/2 or 1/3 or 1/4 duty LCD applications
- Internal time base frequency sources
- Two selectable buzzer frequencies (2kHz/4kHz)
- Power down command reduces power consumption
- Built-in time base generator and WDT
- Time base or WDT overflow output
- 8 kinds of time base/WDT clock sources
- 32×4 LCD driver
- Built-in 32×4 bit display RAM
- 3-wire serial interface
- Internal LCD driving frequency source
- Software configuration feature
- Data mode and command mode instructions
- R/W address auto increment
- Three data accessing modes
- VLCD pin for adjusting LCD operating voltage
- HT1621B: 48-pin SSOP/LQFP packages
  HT1621D: 28-pin SKDIP package
  HT1621G: Gold bumped chip

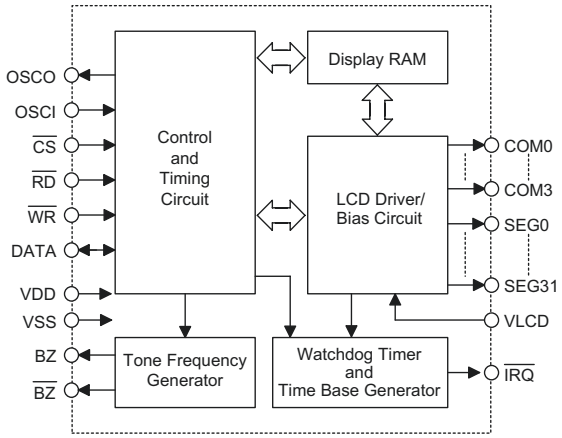
General Description

The HT1621 is a 128 pattern (32×4), memory mapping, and multi-function LCD driver. The S/W configuration feature of the HT1621 makes it suitable for multiple LCD applications including LCD modules and display subsystems. Only three or four lines are required for the interface between the host controller and the HT1621. The HT1621 contains a power down command to reduce power consumption.

Selection Table

HT162X	HT1620	HT1621	HT1622	HT16220	HT1623	HT1625	HT1626
COM	4	**4**	8	8	8	8	16
SEG	32	**32**	32	32	48	64	48
Built-in Osc.	—	√	√	—	√	√	√
Crystal Osc.	√	√	—	√	√	√	√

Block Diagram



Note: $\overline{CS}$: Chip selection

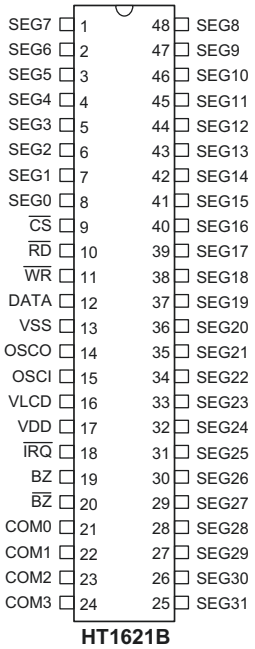
BZ, $\overline{BZ}$: Tone outputs

$\overline{WR}$, $\overline{RD}$, DATA: Serial interface

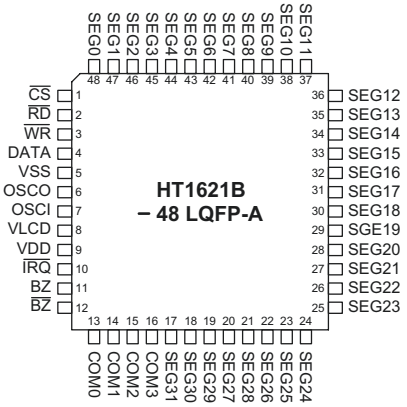
COM0~COM3, SEG0~SEG31: LCD outputs

$\overline{IRQ}$: Time base or WDT overflow output

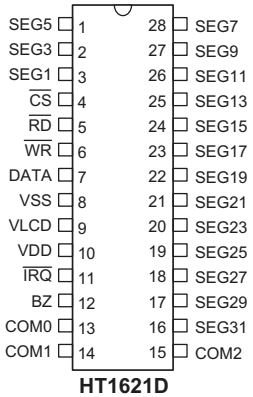
Pin Assignment



HT1621B
– 48 SSOP-A

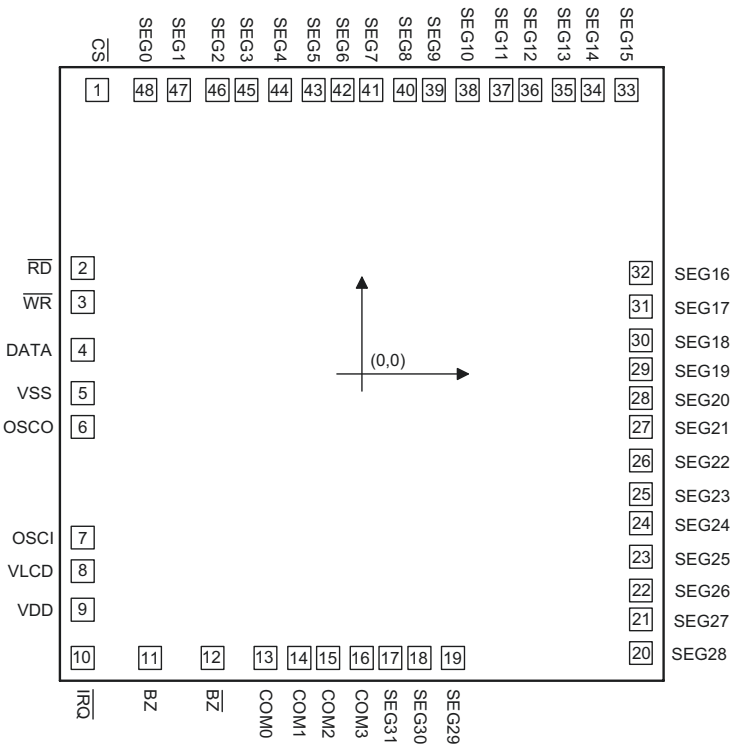


HT1621B
– 48 LQFP-A



HT1621D
– 28 SKDIP-A

Pad Assignment



Chip size: 127 × 131 (mil)^2

Bump height: 18μm ± 3μm

Min. Bump spacing: 72.36μm

Bump size: 96.042 × 96.042μm^2

* The IC substrate should be connected to VDD in the PCB layout artwork.

Pad Coordinates

Unit: mil

Pad No.	X	Y	Pad No.	X	Y
1	–55.04	59.46	25	58.14	–25.29
2	–58.52	22.18	26	58.14	–18.66
3	–58.52	15.56	27	58.14	–11.94
4	–58.52	5.36	28	58.14	–5.31
5	–58.52	–4.51	29	58.14	1.32
6	–58.52	–11.14	30	58.14	7.95
7	–58.52	–34.76	31	58.14	14.58
8	–58.52	–41.90	32	58.14	21.21
9	–58.52	–49.13	33	55.55	59.46
10	–58.52	–59.08	34	48.92	59.46
11	–44.07	–59.08	35	42.29	59.46
12	–31.58	–59.08	36	35.66	59.46
13	–20.70	–59.08	37	29.03	59.46
14	–13.98	–59.08	38	22.40	59.46
15	–7.05	–59.08	39	15.77	59.46
16	–0.34	–59.08	40	9.14	59.46
17	6.33	–59.08	41	2.42	59.46
18	12.96	–59.08	42	–4.21	59.46
19	19.59	–59.08	43	–10.84	59.46
20	58.14	–58.44	44	–17.47	59.46
21	58.14	–51.81	45	–24.10	59.46
22	58.14	–45.18	46	–30.73	59.46
23	58.14	–38.55	47	–38.17	59.46
24	58.14	–31.92	48	–45.39	59.46

Pad Description

Pad No.	Pad Name	I/O	Function
1	$\overline{CS}$	I	Chip selection input with pull-high resistor When the $\overline{CS}$ is logic high, the data and command read from or written to the HT1621 are disabled. The serial interface circuit is also reset. But if $\overline{CS}$ is at logic low level and is input to the $\overline{CS}$ pad, the data and command transmission between the host controller and the HT1621 are all enabled.
2	$\overline{RD}$	I	READ clock input with pull-high resistor Data in the RAM of the HT1621 are clocked out on the falling edge of the $\overline{RD}$ signal. The clocked out data will appear on the DATA line. The host controller can use the next rising edge to latch the clocked out data.
3	$\overline{WR}$	I	WRITE clock input with pull-high resistor Data on the DATA line are latched into the HT1621 on the rising edge of the $\overline{WR}$ signal.
4	DATA	I/O	Serial data input/output with pull-high resistor
5	VSS	—	Negative power supply, ground
7	OSCI	I	The OSCI and OSCO pads are connected to a 32.768kHz crystal in order to generate a system clock. If the system clock comes from an external clock source, the external clock source should be connected to the OSCI pad. But if an on-chip RC oscillator is selected instead, the OSCI and OSCO pads can be left open.
6	OSCO	O	
8	VLCD	I	LCD power input
9	VDD	—	Positive power supply
10	$\overline{IRQ}$	O	Time base or WDT overflow flag, NMOS open drain output
11, 12	BZ, $\overline{BZ}$	O	2kHz or 4kHz tone frequency output pair
13~16	COM0~COM3	O	LCD common outputs
48~17	SEG0~SEG31	O	LCD segment outputs

Absolute Maximum Ratings

Supply Voltage ..... V_{SS}–0.3V to V_{SS}+5.5V

Input Voltage ..... V_{SS}–0.3V to V_{DD}+0.3V

Storage Temperature ..... –50°C to 125°C

Operating Temperature ..... –40°C to 85°C

Note: These are stress ratings only. Stresses exceeding the range specified under "Absolute Maximum Ratings" may cause substantial damage to the device. Functional operation of this device at other conditions beyond those listed in the specification is not implied and prolonged exposure to extreme conditions may affect device reliability.

D.C. Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
V_{DD}	Operating Voltage	—	—	2.4	—	5.2	V
I_{DD1}	Operating Current	3V	No load/LCD ON On-chip RC oscillator	—	150	300	μA
		5V		—	300	600	μA
I_{DD2}	Operating Current	3V	No load/LCD ON Crystal oscillator	—	60	120	μA
		5V		—	120	240	μA
I_{DD3}	Operating Current	3V	No load/LCD ON External clock source	—	100	200	μA
		5V		—	200	400	μA
I_{STB}	Standby Current	3V	No load, Power down mode	—	0.1	5	μA
		5V		—	0.3	10	μA
V_{IL}	Input Low Voltage	3V	DATA, $\overline{WR}$, $\overline{CS}$, $\overline{RD}$	0	—	0.6	V
		5V		0	—	1.0	V
V_{IH}	Input High Voltage	3V	DATA, $\overline{WR}$, $\overline{CS}$, $\overline{RD}$	2.4	—	3.0	V
		5V		4.0	—	5.0	V
I_{OL1}	DATA, BZ, $\overline{BZ}$, $\overline{IRQ}$	3V	V_{OL}=0.3V	0.5	1.2	—	mA
		5V	V_{OL}=0.5V	1.3	2.6	—	mA
I_{OH1}	DATA, BZ, $\overline{BZ}$	3V	V_{OH}=2.7V	–0.4	–0.8	—	mA
		5V	V_{OH}=4.5V	–0.9	–1.8	—	mA
I_{OL2}	LCD Common Sink Current	3V	V_{OL}=0.3V	80	150	—	μA
		5V	V_{OL}=0.5V	150	250	—	μA
I_{OH2}	LCD Common Source Current	3V	V_{OH}=2.7V	–80	–120	—	μA
		5V	V_{OH}=4.5V	–120	–200	—	μA
I_{OL3}	LCD Segment Sink Current	3V	V_{OL}=0.3V	60	120	—	μA
		5V	V_{OL}=0.5V	120	200	—	μA
I_{OH3}	LCD Segment Source Current	3V	V_{OH}=2.7V	–40	–70	—	μA
		5V	V_{OH}=4.5V	–70	–100	—	μA
R_{PH}	Pull-high Resistor	3V	DATA, $\overline{WR}$, $\overline{CS}$, $\overline{RD}$	60	120	200	kΩ
		5V		30	60	100	kΩ

A.C. Characteristics

Ta=25°C

Symbol	Parameter	Test Conditions		Min.	Typ.	Max.	Unit
		V_{DD}	Conditions				
f_{SYS1}	System Clock	—	On-chip RC oscillator	—	256	—	kHz
f_{SYS2}	System Clock	—	Crystal oscillator	—	32.768	—	kHz
f_{SYS3}	System Clock	—	External clock source	—	256	—	kHz
f_{LCD}	LCD Clock	—	On-chip RC oscillator	—	f_{SYS1}/1024	—	Hz
		—	Crystal oscillator	—	f_{SYS2}/128	—	Hz
		—	External clock source	—	f_{SYS3}/1024	—	Hz
t_{COM}	LCD Common Period	—	n: Number of COM	—	n/f_{LCD}	—	s
f_{CLK1}	Serial Data Clock ($\overline{WR}$ pin)	3V	Duty cycle 50%	4	—	150	kHz
		5V		4	—	300	kHz
f_{CLK2}	Serial Data Clock ($\overline{RD}$ pin)	3V	Duty cycle 50%	—	—	75	kHz
		5V		—	—	150	kHz
f_{TONE}	Tone Frequency	—	On-chip RC oscillator	—	2.0 or 4.0	—	kHz
t_{CS}	Serial Interface Reset Pulse Width (Figure 3)	—	$\overline{CS}$	—	250	—	ns
t_{CLK}	$\overline{WR}$, $\overline{RD}$ Input Pulse Width (Figure 1)	3V	Write mode	3.34	—	125	μs
			Read mode	6.67	—	—	
		5V	Write mode	1.67	—	125	μs
			Read mode	3.34	—	—	
t_r, t_f	Rise/Fall Time Serial Data Clock Width (Figure 1)	—	—	—	120	—	ns
t_{su}	Setup Time for DATA to $\overline{WR}$, $\overline{RD}$ Clock Width (Figure 2)	—	—	—	120	—	ns
t_h	Hold Time for DATA to $\overline{WR}$, $\overline{RD}$ Clock Width (Figure 2)	—	—	—	120	—	ns
t_{su1}	Setup Time for $\overline{CS}$ to $\overline{WR}$, $\overline{RD}$ Clock Width (Figure 3)	—	—	—	100	—	ns
t_{h1}	Hold Time for $\overline{CS}$ to $\overline{WR}$, $\overline{RD}$ Clock Width (Figure 3)	—	—	—	100	—	ns

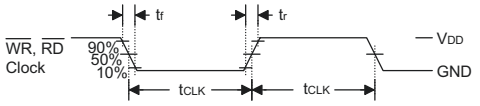


Figure 1

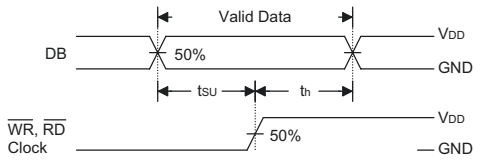


Figure 2

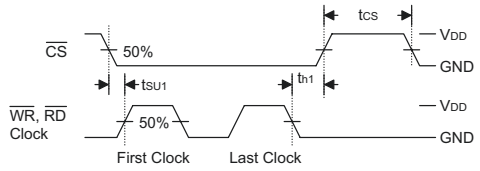


Figure 3

ANEXO VI. CÓDIGO IMPLEMENTADO

Las librerías que se han utilizado son las siguientes:

- ADXL345.h: para la comunicación con el acelerómetro (I2C), desarrollada por SparkFun (https://github.com/sparkfun/SparkFun_ADXL345_Arduino_Library)
- TinyGPS++.h: para la decodificación de los mensajes del módulo GPS, desarrollada por Mikal Hart (http://arduiniana.org/libraries/tinygpsplus/)
- SPI.h: para la comunicación con la tarjeta SD, por Arduino.
- SD.h: para la manipulación de los ficheros en la tarjeta SD, por Arduino.
- HT1621.h: para la gestión de la pantalla LCD, desarrollada por Valerionew (https://github.com/valerionew/ht1621-7-seg)

```
#include <ADXL345.h> //Acelerómetro
#include <TinyGPS++.h> //GPS
#include <SPI.h> //Comunicación con la SD
#include <SD.h> //Ficheros SD
#include <HT1621.h> //LCD

ADXL345 acelerometro;
TinyGPSPlus gps;
File Fichero;
HT1621 lcd;

int x, y, z, acelx, acely, acelz, indice=0, mostrar = 1;
int xReg [600];
int filtrox [8];
int filtroy [8];
int filtroz [8];
int palada [4];
int media, PPM,i;
long t_siguiente, periodo=10, iHD;
boolean anterior = false, cambioHC = false, cambioGPS = false;
int k=15; //Constante para ajustar la sensitividad del dispositivo
byte lectura;
String HCbuffer = "";
double velocidad, mediav;
double filtrov[5];
int xHD, yHD, zHD;


void setup() {
  //LCD
  inicializaLCD();

  //Acel
  inicializaAcelerometro();

  //PC
```

```
  //Serial.begin(115200); //Comunicación serie con el ordenador (Para
debugging)

  //COM
  inicializaComunicaciones(1200);

  //GPS
  inicializaGPS();

  //SD
  compruebaSD();
  abrirFichero("sesion.txt");
}

void loop() {
  t_siguiente=millis()+periodo;

  medirAceleraciones();

  if (paladaDetectada()){
    PPM=calcularRitmo();
    lcd.print(long(PPM)); //La mostramos por pantalla
  }
  actualizarMedia();

  if (indice%100==0){ //una vez cada segundo
    if(cambioGPS){
      guardarSD();
      enviarEntrenador();
      cambioGPS=false;
    }
  }

  // Control de la inclinación
  if (acely>342){
    lcd.setBatteryLevel(3);
  }else{
    lcd.setBatteryLevel(0);
  }

  //Modo TECNICA
  if(cambioHC){
    if(HCbuffer.startsWith("HD")){
      grabarHD();
    }
  }
  cambioHC=false;
  HCbuffer="";
  retrasoHasta (t_siguiente);
}



int calculaMedia600 (int Reg [600]) {
```

```
  media = 0;
  for (int i = 0; i < 600; i++) {
    media = media + xReg[i];
  }
  return media / 600;
}

static void retrasoHasta(unsigned long ms) {
  int g=0;
  do {
    while (Serial3.available()){ //Si hay datos disponibles por el
puerto serie del GPS
      gps.encode(Serial3.read()); //Guarda en un buffer la información
recibida
      cambioGPS = true;
      //Serial.println("GPS");
    }
    while (Serial1.available()){
      i++;
      lectura = Serial1.read();
      HCbuffer = char(lectura);
      cambioHC = true;
    }
  }while (millis() < ms);
}

static void inicializaLCD(){
  lcd.begin(45, 44, 43, 42); // (cs, wr, Data, LED)
  digitalWrite(42, HIGH);
  lcd.clear();
}

static void inicializaAcelerometro(){
  acelerometro.powerOn();
}

static void inicializaComunicaciones(int velocidad){
  Serial1.begin(velocidad);
}

static void inicializaGPS(){
  Serial3.begin(9600);
}

static void compruebaSD(){
  if (!SD.begin(53)) {
    Serial1.println("fallo SD");
  }
}

static boolean abrirFichero(String nombre){
  Fichero = SD.open(nombre, FILE_WRITE);
  if (!Fichero) {
    //Serial.println("¡¡Archivo falló!!");
```

```
    Serial1.println("fallo Archivo");
    return false;
  }else return true;
}

static void medirAceleraciones(){
  acelerometro.readAccel(&x, &y, &z); //leemos los valores del
acelerómetro y las guardamos en x,y,z
  acelx=0;
  acely=0;
  acelz=0;
  for (int i=7;i>0;i--){
    filtrox[i] = filtrox[i-1];
    acelx+=filtrox[i];
    filtroy[i] = filtroy[i-1];
    acely+=filtroy[i];
    filtroz[i] = filtroz[i-1];
    acelz+=filtroz[i];
  }
  filtrox[0] = x;
  acelx = (acelx+filtrox[0]) / 8;
  filtroy[0] = y+14;
  acely = (acely+filtroy[0]) / 8;
  filtroz[0] = z-240;
  acelz = (acelz+filtroz[0]) / 8;
}

static boolean paladaDetectada(){
  boolean aux=false;
  if (acelx > (media + k)) {
    if (anterior == false) { //palada detectada
      aux=true;
    }
    anterior = true;
  } else {
    anterior = false;
  }
  return aux;
}

static int calcularRitmo(){
  palada[3] = palada[2];
  palada[2] = palada[1];
  palada[1] = palada[0];
  palada[0] = millis();
  //Calculamos el ritmo de paladas
  return (3 * 60000) / (palada[0] - palada[3]); //3
paladas*(60000ms=1min)/t(ms)
}

static void actualizarMedia(){
  xReg[indice] = acelx;
  indice = (indice + 1) % 600;
  media = calculaMedia600(xReg);
```

```
}

static void guardarSD(){
  if(Fichero){
    //Serial.println("Guardando");
    String bufferDatos="";
    if (gps.time.isValid()){
      digitalWrite(LED_BUILTIN, HIGH);
      if (gps.time.hour() < 10) bufferDatos+="0";
      bufferDatos+=gps.time.hour();
      bufferDatos+=":";
      if (gps.time.minute() < 10) bufferDatos+="0";
      bufferDatos+=gps.time.minute();
      bufferDatos+=":";
      if (gps.time.second() < 10) bufferDatos+="0";
      bufferDatos+=gps.time.second();
    }else{
      bufferDatos+="INVALID";
    }
    bufferDatos+=",";
    if (gps.location.isValid()){
      bufferDatos+=String(gps.location.lat(), 6);
      bufferDatos+=",";
      bufferDatos+=String(gps.location.lng(), 6);
      bufferDatos+=",";
      bufferDatos+=String(gps.speed.kmph(),1);
    }else{
      bufferDatos+="INVALID";
    }
    bufferDatos+=",";
    bufferDatos+=PPM;
    Fichero.println(bufferDatos);
    Fichero.flush();
  }
}

static void enviarEntrenador(){
  String bufferDatos="";
  if (gps.time.isValid()){
    digitalWrite(LED_BUILTIN, HIGH);
    if (gps.time.hour() < 10) bufferDatos+="0";
    bufferDatos+=gps.time.hour();
    bufferDatos+=":";
    if (gps.time.minute() < 10) bufferDatos+="0";
    bufferDatos+=gps.time.minute();
    bufferDatos+=":";
    if (gps.time.second() < 10) bufferDatos+="0";
    bufferDatos+=gps.time.second();
  }else{
    bufferDatos+="INV";
  }
  bufferDatos+=",";
  if (gps.location.isValid()){
    bufferDatos+=String(gps.speed.kmph(),1);
```

```
  }else{
    bufferDatos+="INV";
  }
  bufferDatos+=",";
  bufferDatos+=PPM;
  Serial1.println(bufferDatos);
}

static void grabarHD(){
  Fichero.close();
  if (!abrirFichero("HD.txt")) {
    Serial1.println("¡¡ArchivoHD falló!!");
  } else {
    //Empieza el modo HD
    Serial1.println("HD");
    //Apagamos los módulos no utilizados
    Serial3.end();
    Serial1.end();
    lcd.clear();

    //aumentamos la velocidad de muestreo del acelerómetro
    Wire.setClock(400000);
    acelerometro.setRate(800);

    String bufferDatos;
    long t_inicio=millis();
    long t_fin=t_inicio+10000;
    long t_siguiente;

    while (millis()<t_fin){
      t_siguiente=micros()+1250;
      acelerometro.readAccel(&xHD, &yHD, &zHD);
      bufferDatos="";
      bufferDatos+=(millis()-t_inicio);
      bufferDatos+=",";
      bufferDatos+=xHD;
      bufferDatos+=",";
      bufferDatos+=yHD;
      bufferDatos+=",";
      bufferDatos+=zHD;
      Fichero.println(bufferDatos);
      delayMicroseconds(t_siguiente-micros());
    }
    inicializaComunicaciones(1200);
    inicializaGPS();
    //Volvemos el acelerómetro al modo inicial
    acelerometro.setRate(100);
    Wire.setClock(100000);
    Serial1.println(iHD);
    Fichero.close();
    abrirFichero("sesion.txt");
  }
}
```