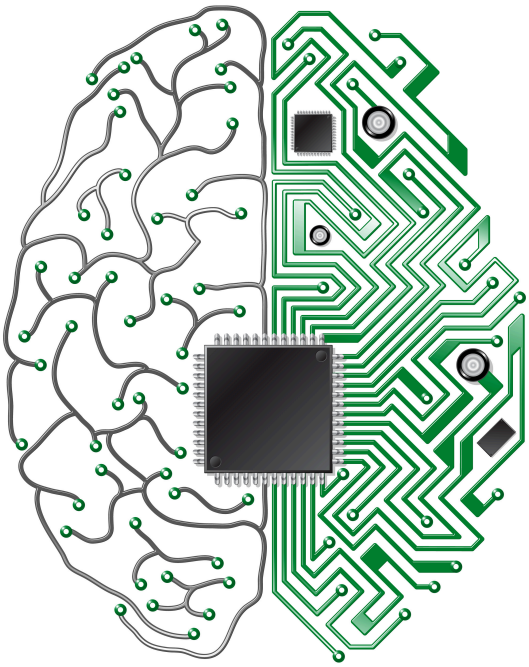
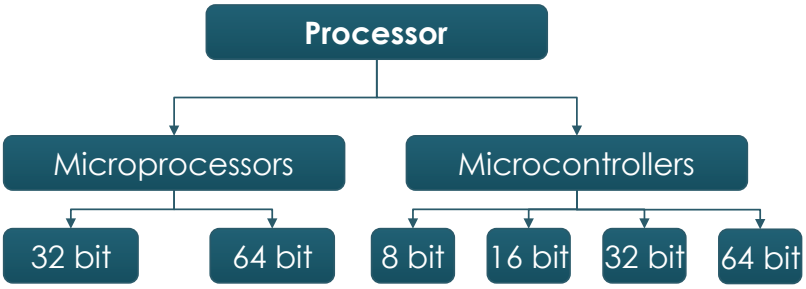


What makes smart devices devices Smart?

Processor

iRoboticist.com

Processor : Types



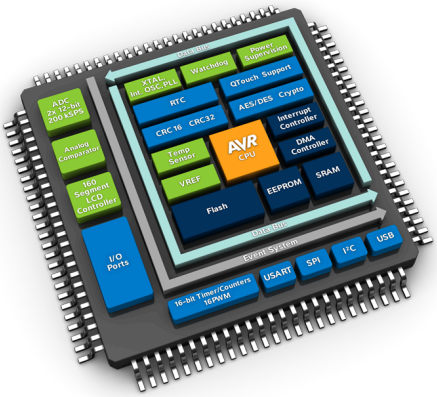
uC Manufactures

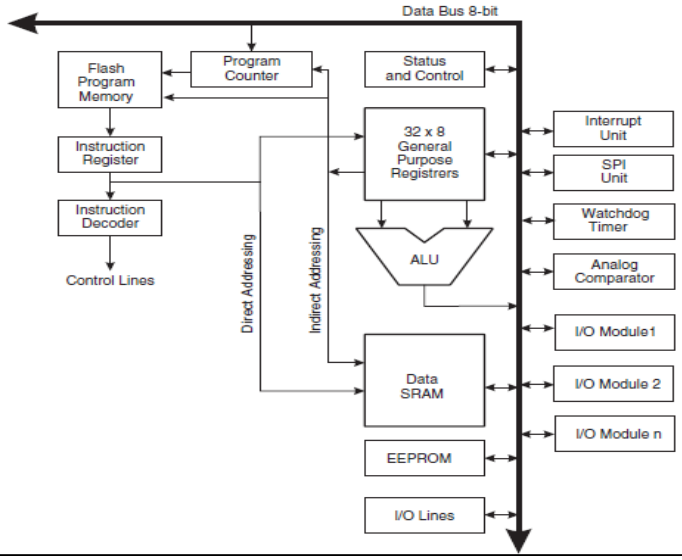
- Atmel – AVR atmega32u4
- Microchip - PIC
- NXP
- Texas Instruments
- Analog Devices
- Intel
- Hitachi
- Freescale
- Silicon Labs

Why AVR ??

- Based on the popular Harvard Architecture
  - Separate memories and buses for program and data
  - Program Flash memory space is divided in two sections, the Boot Program section and the Application Program section.
- Huge support
- Easy to use and program

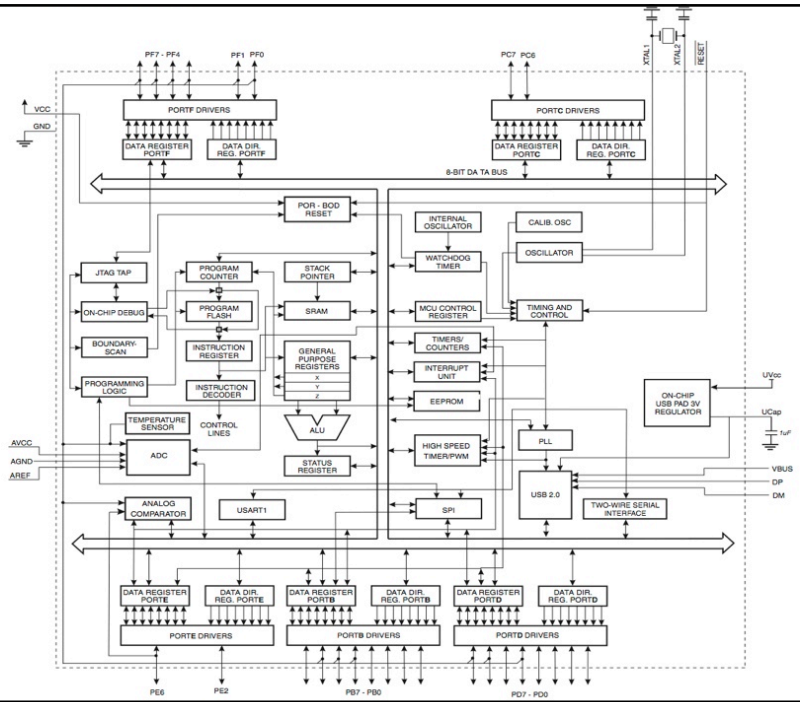
AVR Microcontroller Architecture





ATmega32u4 Block Diagram

Flash: 32 Kbytes
Max. Operating Freq.: 16 MHz
CPU: 8-bit AVR
I/O Pins: 26
Ext Interrupts: 13
SPI: 2
TWI (I2C): 1
UART: 1
ADC channels:12
SRAM: 2.5 Kbytes
EEPROM (Bytes):1024
Operating Voltage (Vcc): 2.7 to 5.5
PWM Channels: 8



Atmel Atmega32u4 : 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
  - Up to 16 MIPS Throughput at 16 MHz
- **Non-volatile Program and Data Memories**
  - 32K Bytes of In-System Self-Programmable Flash 2.5K Bytes Internal SRAM
  - 1K Bytes Internal EEPROM – Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - In-System Programming by On-chip Boot Program
- **USB 2.0 Full-speed/Low Speed Device Module with Interrupt on Transfer Completion**

Peripheral Features

- **Timer / Counter**
  - On-chip PLL for USB and High Speed Timer: 32 up to 96 MHz operation
  - One 8-bit Timer/Counter with Separate Prescaler and Compare Mode
  - Two 16-bit Timer/Counter with Separate Prescaler, Compare- and Capture Mode
  - One 10-bit High-Speed Timer/Counter with PLL (64 MHz) and Compare Mode
- **PWM**
  - Four 8-bit **PWM** Channels
  - Four **PWM** Channels with Programmable Resolution from 2 to 16 Bits
  - Six **PWM** Channels for High Speed Operation, with Programmable Resolution from 2 to 11 Bits
- **ADC**
  - 12-channels, 10-bit ADC
- **Serial Interface**
  - Programmable Serial **USART** with Hardware Flow Control
  - Master/Slave **SPI** Serial Interface
  - Byte Oriented 2-wire Serial Interface

iRoboticist.com

Arduino

- **Arduino** is a **tool** for making computers that can sense and control more of the physical world than your desktop computer.
- It's an open-source physical computing **platform based on a simple microcontroller board**, and **a development environment** for writing software for the board.

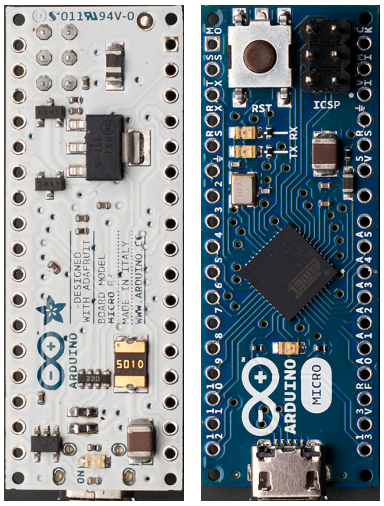
iRoboticist.com

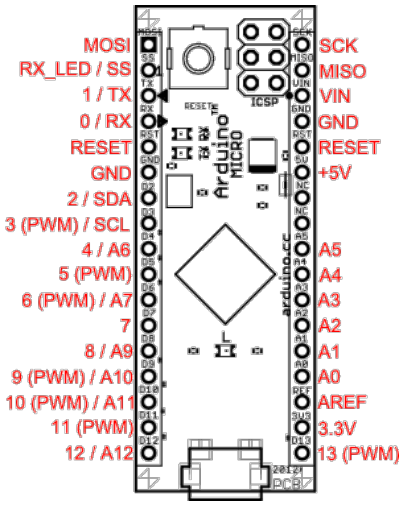
Arduino Advantage

- Arduino takes the messy details of microcontroller programming and wrap it up in an easy-to-use package. It simplifies the process of working with microcontrollers.
  - **Inexpensive**
  - **Cross-platform**
    - Windows
    - Macintosh OSX and
    - Linux
  - **Simple, clear programming environment**
    - easy-to-use for beginners
    - flexible for advanced users
  - **Open source and extensible software & hardware**
  - **Multiple Hardware option**
    - Arduino's are available in multiple form factors, speed and I/O requirements.

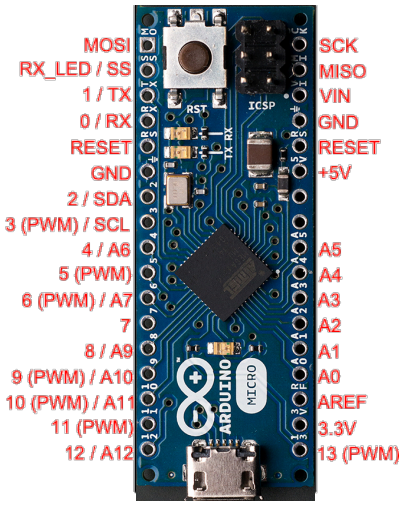
iRoboticist.com

Arduino Micro - atmega32u4

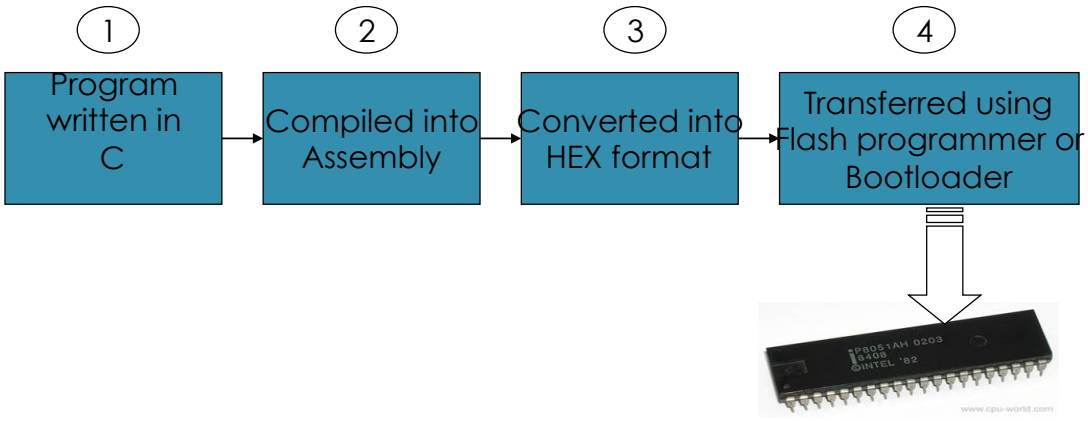




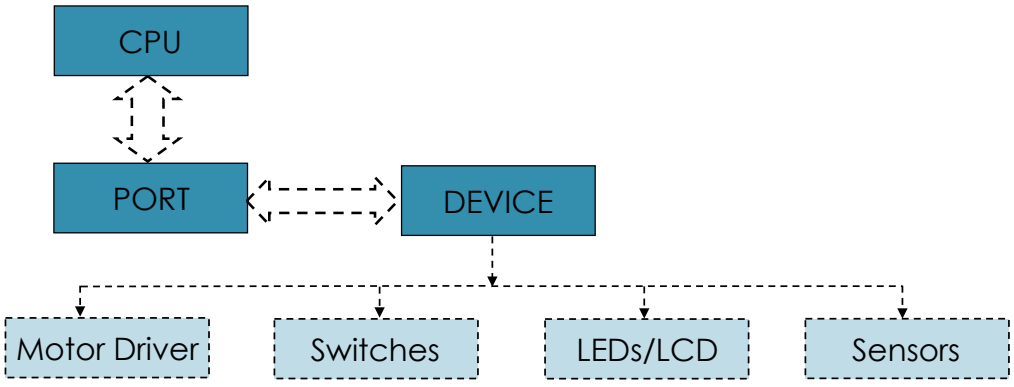
Pin Multiplexing:



Programming : The Process

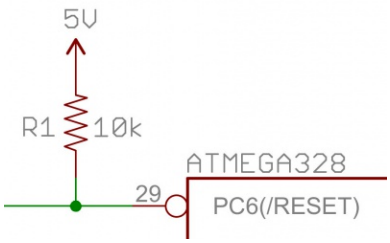


I/O Ports:



Floating & Pull-up Resistor

- For a pin configured as an **input**, if there is nothing connected to the pin and your program reads the state of the pin, will it be **high** or **low**? It is difficult to tell. This phenomena is referred to as **floating**.
- To prevent this unknown state, a pull-up or pull-down resistor will insure that the pin is in either a high or low state, while also using a low amount of current.



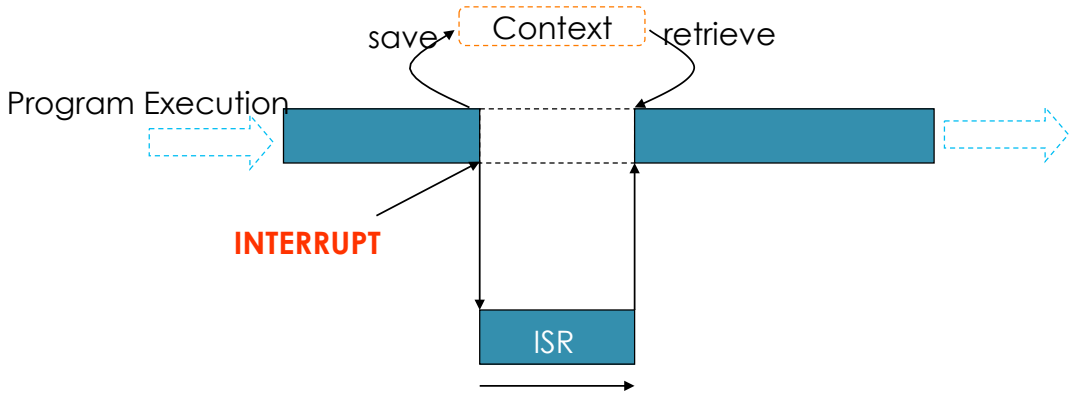
Serial Communication: UART

UART- Universal Asynchronous Receiver Transmitter

Interrupts

- Asynchronous input which changes the normal flow of execution of the program
- Higher Priority than program

Interrupts Service Routine - ISR



Types Of Interrupts:

Maskable

Serial
Ext. Interrupt
etc

OR

Non Maskable

RESET

Hardware

Ext. Interrupt
RESET
Timer/Counter
etc.

Software

Divide by zero

Setup, Interrupts

```
void setup ( ) {
attachInterrupt (interrupt, function, mode);
}
```

- You can designate an interrupt function to Arduino Micro pins 0(RX), 1(TX), 2 and 3
- This is a way around the linear processing of Arduino

Setup, Interrupts

void setup ( ) { attachInterrupt (interrupt, function, mode) }

- **Interrupt:** the number of the interrupt, 0 or 1, corresponding to Arduino pins # 2 and 3 respectively
- **Function:** the function to call when the interrupt occurs
- **Mode:** defines when the interrupt should be triggered

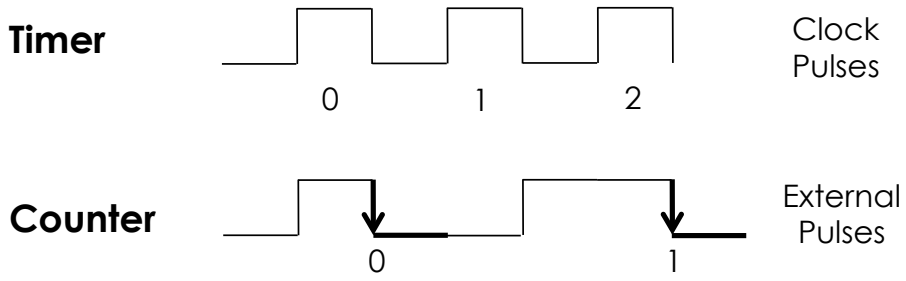
Setup, Interrupts

void setup ( ) { attachInterrupt (interrupt, function, mode) }

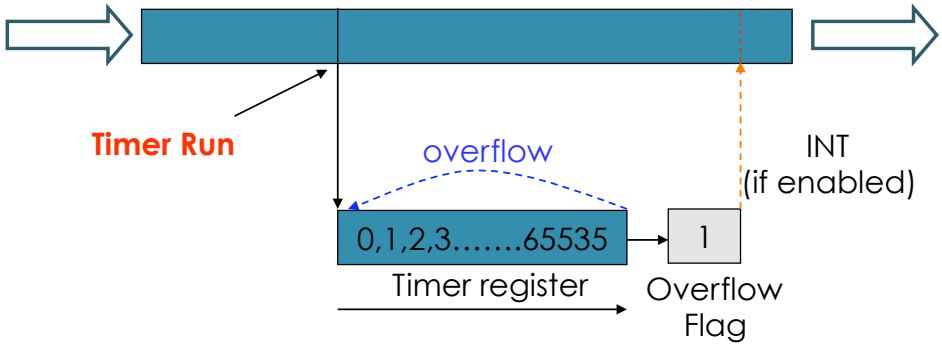
LOW whenever pin state is low
CHANGE whenever pin changes value
RISING whenever pin goes from low to high
FALLING whenever pin goes from low to high

Don't forget to CAPITALIZE

Timer / Counter



Timers / Counters: Program Execution



Modes: 8 bit, 16 bit,10 bit timers