



Tips4FYP

Anthony Faustine, Kelvin Paul and Garimo Paul

Outline

1. Building Blocks of the Electronics Systems
2. MCU Basics
3. Arduino Platform
4. Prototyping Circuits
5. Pull-Up and Pull-Down Resistors
6. Embedded Devices
7. Designing Your Project

Electronics Systems Block

Most of electronics systems consists of the following blocks:

- Control Unit
- Sensors and or Actuators
- Communications module
- Power unit

Electronics Systems Block

Most of electronics systems consists of the following blocks:

- Control Unit
- Sensors and or Actuators
- Communications module
- Power unit

Electronics Systems Block

Most of electronics systems consists of the following blocks:

- Control Unit
- Sensors and or Actuators
- Communications module
- Power unit

Electronics Systems Block

Most of electronics systems consists of the following blocks:

- Control Unit
- Sensors and or Actuators
- Communications module
- Power unit

Control Units

Electronics system utilize micro-controller as the main control unit.

- A microcontroller is a small computer in a single integrated circuit.
- It contain a processor core, a memory, and programmable I/O peripheral.
- MCU
 - The 'brain' controls everything
 - Reads input from sensors
 - Drives outputs
 - LED, Switch, Motor
 - Communicates!

Control Units

Electronics system utilize micro-controller as the main control unit.

- A microcontroller is a small computer in a single integrated circuit.
- It contain a processor core, a memory, and programmable I/O peripheral.
- MCU
 - The 'brain' controls everything
 - Reads input from sensors
 - Drives outputs
 - LED, Switch, Motor
 - Communicates!

Control Units

Electronics system utilize micro-controller as the main control unit.

- A microcontroller is a small computer in a single integrated circuit.
- It contain a processor core, a memory, and programmable I/O peripheral.
- MCU
 - The 'brain' controls everything
 - Reads input from sensors
 - Drives outputs
 - LED, Switch, Motor
 - Communicates!

Control Units

Electronics system utilize micro-controller as the main control unit.

- A microcontroller is a small computer in a single integrated circuit.
- It contain a processor core, a memory, and programmable I/O peripheral.
- MCU
 - The ‘brain’ controls everything
 - Reads input from sensors
 - Drives outputs
 - LED, Switch, Motor
 - Communicates!

Control Units

Electronics system utilize micro-controller as the main control unit.

- A microcontroller is a small computer in a single integrated circuit.
- It contain a processor core, a memory, and programmable I/O peripheral.
- MCU
 - The ‘brain’ controls everything
 - Reads input from sensors
 - Drives outputs
 - LED, Switch, Motor
 - Communicates!

Control Units

Electronics system utilize micro-controller as the main control unit.

- A microcontroller is a small computer in a single integrated circuit.
- It contain a processor core, a memory, and programmable I/O peripheral.
- MCU
 - The 'brain' controls everything
 - Reads input from sensors
 - Drives outputs
 - LED, Switch, Motor
 - Communicates!

Control Units

Electronics system utilize micro-controller as the main control unit.

- A microcontroller is a small computer in a single integrated circuit.
- It contain a processor core, a memory, and programmable I/O peripheral.
- MCU
 - The 'brain' controls everything
 - Reads input from sensors
 - Drives outputs
 - LED, Switch, Motor
 - Communicates!

Control Units

Electronics system utilize micro-controller as the main control unit.

- A microcontroller is a small computer in a single integrated circuit.
- It contain a processor core, a memory, and programmable I/O peripheral.
- MCU
 - The ‘brain’ controls everything
 - Reads input from sensors
 - Drives outputs
 - LED, Switch, Motor
 - Communicates!

Sensors and Actuators

Sensors: Device that can sense the physical quantities and convert into signal which can be interpreted by the MCU

- Fall into two types: Analog and Digital Sensor
- A good sensor obeys the following rules:



Sensors and Actuators

Sensors: Device that can sense the physical quantities and convert into signal which can be interpreted by the MCU

- Fall into two types: Analog and Digital Sensor
- A good sensor obeys the following rules:



Sensors and Actuators

Sensors: Device that can sense the physical quantities and convert into signal which can be interpreted by the MCU

- Fall into two types: Analog and Digital Sensor
- A good sensor obeys the following rules:

- Is sensitive to the measured property only
- Is insensitive to any other property likely to be encountered in its application
- Does not influence the measured property



Sensors and Actuators

Sensors: Device that can sense the physical quantities and convert into signal which can be interpreted by the MCU

- Fall into two types: Analog and Digital Sensor
- A good sensor obeys the following rules:
 - Is sensitive to the measured property only
 - Is insensitive to any other property likely to be encountered in its application
 - Does not influence the measured property



Sensors and Actuators

Sensors: Device that can sense the physical quantities and convert into signal which can be interpreted by the MCU

- Fall into two types: Analog and Digital Sensor
- A good sensor obeys the following rules:
 - Is sensitive to the measured property only
 - Is insensitive to any other property likely to be encountered in its application
 - Does not influence the measured property



Sensors and Actuators

Sensors: Device that can sense the physical quantities and convert into signal which can be interpreted by the MCU

- Fall into two types: Analog and Digital Sensor
- A good sensor obeys the following rules:
 - Is sensitive to the measured property only
 - Is insensitive to any other property likely to be encountered in its application
 - Does not influence the measured property



Sensors and Actuators

Sensors: Device that can sense the physical quantities and convert into signal which can be interpreted by the MCU

- Fall into two types: Analog and Digital Sensor
- A good sensor obeys the following rules:
 - Is sensitive to the measured property only
 - Is insensitive to any other property likely to be encountered in its application
 - Does not influence the measured property



Sensors and Actuators

Sensors: Device that can sense the physical quantities and convert into signal which can be interpreted by the MCU

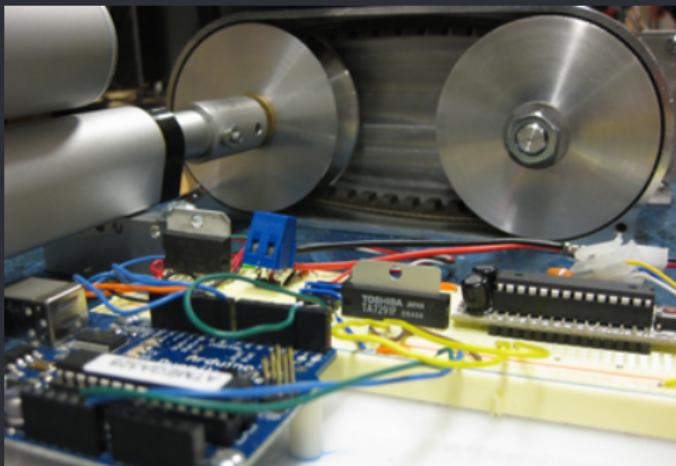
- Fall into two types: Analog and Digital Sensor
- A good sensor obeys the following rules:
 - Is sensitive to the measured property only
 - Is insensitive to any other property likely to be encountered in its application
 - Does not influence the measured property



Sensors and Actuators

Actuators: Device that converts signals to corresponding physical action.

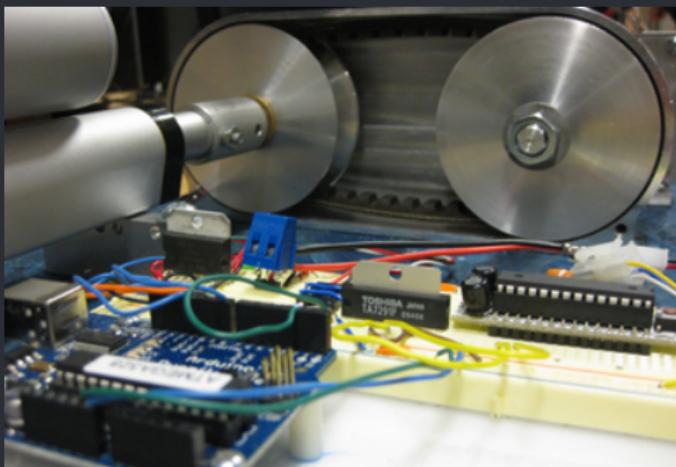
- Actuator acts as an output device
- If the system is designed for monitoring purpose only ⇒ no need for including an actuator.



Sensors and Actuators

Actuators: Device that converts signals to corresponding physical action.

- Actuator acts as an output device
- If the system is designed for monitoring purpose only \Rightarrow no need for including an actuator.



Communication Interface

Actuators: Device that provides interaction with various subsystems and the external world.

- Provide connectivity between devices and the Internet
- Two different perspectives:
 1. On board communication: Serial, I2C, i-Wire, SPI etc
 2. External communication: IR, Wi-Fi, Ethernet, Blue-tooth, GSM(GPRS), RF, ZigBee etc.

Communication Interface

Actuators: Device that provides interaction with various subsystems and the external world.

- Provide connectivity between devices and the Internet
- Two different perspectives:
 1. On board communication: Serial, I2C, i-Wire, SPI etc
 2. External communication: IR, Wi-Fi, Ethernet, Blue-tooth, GSM(GPRS), RF, ZigBee etc.

Communication Interface

Actuators: Device that provides interaction with various subsystems and the external world.

- Provide connectivity between devices and the Internet
- Two different perspectives:
 1. On board communication: Serial, I2C, i-Wire, SPI etc
 2. External communication: IR, Wi-Fi, Ethernet, Blue-tooth, GSM(GPRS), RF, ZigBee etc.

Communication Interface

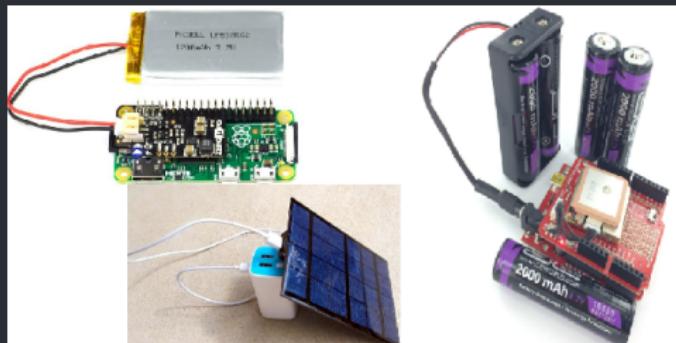
Actuators: Device that provides interaction with various subsystems and the external world.

- Provide connectivity between devices and the Internet
- Two different perspectives:
 1. On board communication: Serial, I2C, i-Wire, SPI etc
 2. External communication: IR, Wi-Fi, Ethernet, Blue-tooth, GSM(GPRS), RF, ZigBee etc.

Power Units

There are various ways to power electronic systems

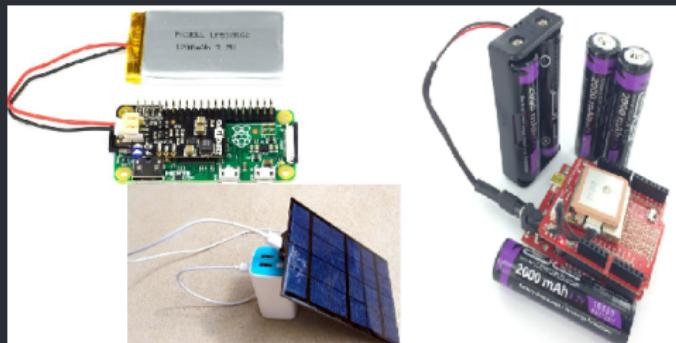
- AC to DC power supplies
- Batteries
- Energy harvesting (Solar, Wind etc)



Power Units

There are various ways to power electronic systems

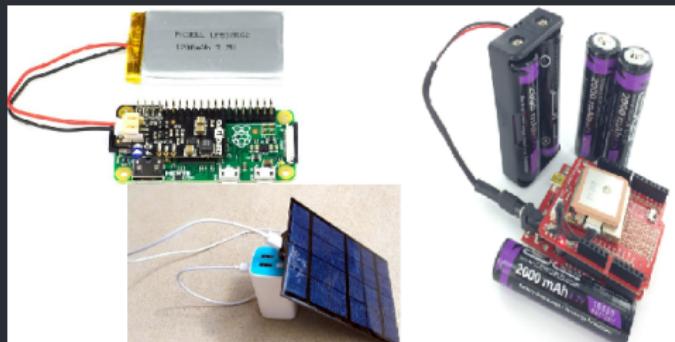
- AC to DC power supplies
- Batteries
- Energy harvesting (Solar, Wind etc)



Power Units

There are various ways to power electronic systems

- AC to DC power supplies
- Batteries
- Energy harvesting (Solar, Wind etc)



Outline

1. Building Blocks of the Electronics Systems
2. MCU Basics
3. Arduino Platform
4. Prototyping Circuits
5. Pull-Up and Pull-Down Resistors
6. Embedded Devices
7. Designing Your Project

Basics of MCU

MCU: Very common component in modern electronics systems.

The main components of MCU

- CPU ⇒ Main processing unit.
- Memory ⇒ Include the program that is being executed and is also available for storing.
- I/O peripheral ⇒ Pins that collect and generate digital signals to other circuit.
- Serial line (TX/RX) ⇒ Allow serial data to be transmitted to or from the MCU.
- A/D converters ⇒ To allow MCU receive analog data for processing
- Timers ⇒ To allow MCU to perform task for certain time period

Basics of MCU

MCU: Very common component in modern electronics systems.

The main components of MCU

- CPU ⇒ Main processing unit.
- Memory ⇒ Include the program that is being executed and is also available for storing.
- I/O peripheral ⇒ Pins that collect and generate digital signals to other circuit.
- Serial line (TX/RX) ⇒ Allow serial data to be transmitted to or from the MCU.
- A/D converters ⇒ To allow MCU receive analog data for processing
- Timers ⇒ To allow MCU to perform task for certain time period

Basics of MCU

MCU: Very common component in modern electronics systems.

The main components of MCU

- CPU ⇒ Main processing unit.
- Memory ⇒ Include the program that is being executed and is also available for storing.
- I/O peripheral ⇒ Pins that collect and generate digital signals to other circuit.
- Serial line (TX/RX) ⇒ Allow serial data to be transmitted to or from the MCU.
- A/D converters ⇒ To allow MCU receive analog data for processing
- Timers ⇒ To allow MCU to perform task for certain time period

Basics of MCU

MCU: Very common component in modern electronics systems.

The main components of MCU

- CPU ⇒ Main processing unit.
- Memory ⇒ Include the program that is being executed and is also available for storing.
- I/O peripheral ⇒ Pins that collect and generate digital signals to other circuit.
- Serial line (TX/RX) ⇒ Allow serial data to be transmitted to or from the MCU.
- A/D converters ⇒ To allow MCU receive analog data for processing
- Timers ⇒ To allow MCU to perform task for certain time period

Basics of MCU

MCU: Very common component in modern electronics systems.

The main components of MCU

- CPU ⇒ Main processing unit.
- Memory ⇒ Include the program that is being executed and is also available for storing.
- I/O peripheral ⇒ Pins that collect and generate digital signals to other circuit.
- Serial line (TX/RX) ⇒ Allow serial data to be transmitted to or from the MCU.
- A/D converters ⇒ To allow MCU receive analog data for processing
- Timers ⇒ To allow MCU to perform task for certain time period

Basics of MCU

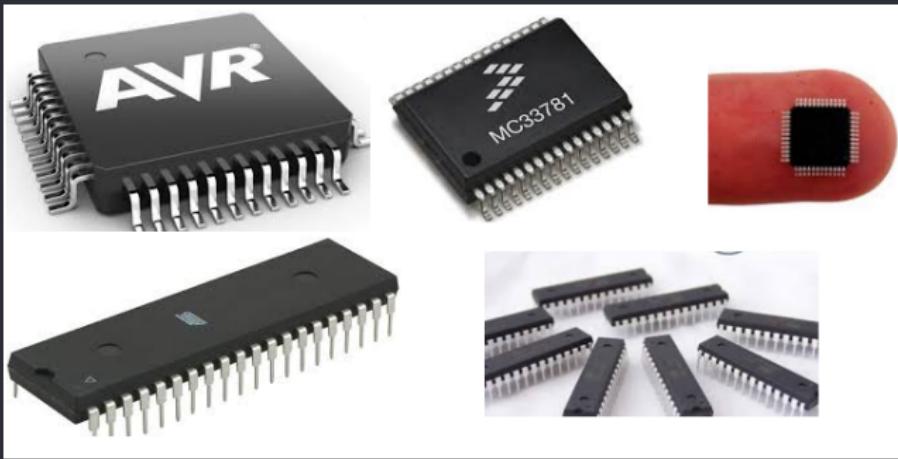
MCU: Very common component in modern electronics systems.

The main components of MCU

- CPU ⇒ Main processing unit.
- Memory ⇒ Include the program that is being executed and is also available for storing.
- I/O peripheral ⇒ Pins that collect and generate digital signals to other circuit.
- Serial line (TX/RX) ⇒ Allow serial data to be transmitted to or from the MCU.
- A/D converters ⇒ To allow MCU receive analog data for processing
- Timers ⇒ To allow MCU to perform task for certain time period

Basics of MCU

MCU: Very common component in modern electronics systems.



Example of MCU

- **PIC:** from Microchip

- Very simple, very proven but low community support.
- It lacks many of the features that other mfg's are building into their chips

- **AVR:** Micro controllers from Atmel.

- They do everything a PIC does,
- Cheap , large number of library files , used in many robotic applications.
- Best for the beginners.
- It is better, faster, cheaper, and simpler.

- **ARM:** UK based company.

- Very powerful and fast, very low-cost but complex
- Large number of timer and ADC's.

Example of MCU

- **PIC:** from Microchip
 - Very simple, very proven but low community support.
 - It lacks many of the features that other mfg's are building into their chips
- **AVR:** Micro controllers from Atmel.
 - They do everything a PIC does,
 - Cheap , large number of library files , used in many robotic applications.
 - Best for the beginners.
 - It is better, faster, cheaper, and simpler.
- **ARM:** UK based company.
 - Very powerful and fast, very low-cost but complex
 - Large number of timer and ADC's.

Example of MCU

- **PIC:** from Microchip
 - Very simple, very proven but low community support.
 - It lacks many of the features that other mfg's are building into their chips
- **AVR:** Micro controllers from Atmel.
 - They do everything a PIC does,
 - Cheap , large number of library files , used in many robotic applications.
 - Best for the beginners.
 - It is better, faster, cheaper, and simpler.
- **ARM:** UK based company.
 - Very powerful and fast, very low-cost but complex
 - Large number of timer and ADC's.

Example of MCU

- **PIC:** from Microchip
 - Very simple, very proven but low community support.
 - It lacks many of the features that other mfg's are building into their chips
- **AVR:** Micro controllers from Atmel.
 - They do everything a PIC does,
 - Cheap , large number of library files , used in many robotic applications.
 - Best for the beginners.
 - It is better, faster, cheaper, and simpler.
- **ARM:** UK based company.
 - Very powerful and fast, very low-cost but complex
 - Large number of timer and ADC's.

Example of MCU

- **PIC:** from Microchip
 - Very simple, very proven but low community support.
 - It lacks many of the features that other mfg's are building into their chips
- **AVR:** Micro controllers from Atmel.
 - They do everything a PIC does,
 - Cheap , large number of library files , used in many robotic applications.
 - Best for the beginners.
 - It is better, faster, cheaper, and simpler.
- **ARM:** UK based company.
 - Very powerful and fast, very low-cost but complex
 - Large number of timer and ADC's.

Example of MCU

- **PIC:** from Microchip
 - Very simple, very proven but low community support.
 - It lacks many of the features that other mfg's are building into their chips
- **AVR:** Micro controllers from Atmel.
 - They do everything a PIC does,
 - Cheap , large number of library files , used in many robotic applications.
 - Best for the beginners.
 - It is better, faster, cheaper, and simpler.
- **ARM:** UK based company.
 - Very powerful and fast, very low-cost but complex
 - Large number of timer and ADC's.

Example of MCU

- **PIC:** from Microchip
 - Very simple, very proven but low community support.
 - It lacks many of the features that other mfg's are building into their chips
- **AVR:** Micro controllers from Atmel.
 - They do everything a PIC does,
 - Cheap , large number of library files , used in many robotic applications.
 - Best for the beginners.
 - It is better, faster, cheaper, and simpler.
- **ARM:** UK based company.
 - Very powerful and fast, very low-cost but complex
 - Large number of timer and ADC's.

Example of MCU

- **PIC:** from Microchip
 - Very simple, very proven but low community support.
 - It lacks many of the features that other mfg's are building into their chips
- **AVR:** Micro controllers from Atmel.
 - They do everything a PIC does,
 - Cheap , large number of library files , used in many robotic applications.
 - Best for the beginners.
 - It is better, faster, cheaper, and simpler.
- **ARM:** UK based company.
 - Very powerful and fast, very low-cost but complex
 - Large number of timer and ADC's.

Example of MCU

- **PIC:** from Microchip
 - Very simple, very proven but low community support.
 - It lacks many of the features that other mfg's are building into their chips
- **AVR:** Micro controllers from Atmel.
 - They do everything a PIC does,
 - Cheap , large number of library files , used in many robotic applications.
 - Best for the beginners.
 - It is better, faster, cheaper, and simpler.
- **ARM:** UK based company.
 - Very powerful and fast, very low-cost but complex
 - Large number of timer and ADC's.

Example of MCU

- **PIC:** from Microchip
 - Very simple, very proven but low community support.
 - It lacks many of the features that other mfg's are building into their chips
- **AVR:** Micro controllers from Atmel.
 - They do everything a PIC does,
 - Cheap , large number of library files , used in many robotic applications.
 - Best for the beginners.
 - It is better, faster, cheaper, and simpler.
- **ARM:** UK based company.
 - Very powerful and fast, very low-cost but complex
 - Large number of timer and ADC's.

Example of MCU

- **PIC:** from Microchip
 - Very simple, very proven but low community support.
 - It lacks many of the features that other mfg's are building into their chips
- **AVR:** Micro controllers from Atmel.
 - They do everything a PIC does,
 - Cheap , large number of library files , used in many robotic applications.
 - Best for the beginners.
 - It is better, faster, cheaper, and simpler.
- **ARM:** UK based company.
 - Very powerful and fast, very low-cost but complex
 - Large number of timer and ADC's.

Example of MCU

- **MSP:** from Texas Instruments.
 - Excel at low-power and portable applications.
 - It lacks many of the features that other mfg's are building into their chips
- **8051:** Was created by Intel.
 - Very old tech ⇒ Developed by Intel in the 1980s,
 - Seems to be the instruction set they love to teach you in college,

Example of MCU

- **MSP:** from Texas Instruments.
 - Excel at low-power and portable applications.
 - It lacks many of the features that other mfg's are building into their chips
- **8051:** Was created by Intel.
 - Very old tech ⇒ Developed by Intel in the 1980s,
 - Seems to be the instruction set they love to teach you in college,

Example of MCU

- **MSP:** from Texas Instruments.
 - Excel at low-power and portable applications.
 - It lacks many of the features that other mfg's are building into their chips
- **8051:** Was created by Intel.
 - Very old tech ⇒ Developed by Intel in the 1980s,
 - Seems to be the instruction set they love to teach you in college,

Example of MCU

- **MSP:** from Texas Instruments.
 - Excel at low-power and portable applications.
 - It lacks many of the features that other mfg's are building into their chips
- **8051:** Was created by Intel.
 - Very old tech ⇒ Developed by Intel in the 1980s.
 - Seems to be the instruction set they love to teach you in college.

Example of MCU

- **MSP:** from Texas Instruments.
 - Excel at low-power and portable applications.
 - It lacks many of the features that other mfg's are building into their chips
- **8051:** Was created by Intel.
 - Very old tech ⇒ Developed by Intel in the 1980s.
 - Seems to be the instruction set they love to teach you in college.

Example of MCU

- **MSP:** from Texas Instruments.
 - Excel at low-power and portable applications.
 - It lacks many of the features that other mfg's are building into their chips
- **8051:** Was created by Intel.
 - Very old tech ⇒ Developed by Intel in the 1980s.
 - Seems to be the instruction set they love to teach you in college.

Programming MCU

The main function of the MCU is to control things through I/O interfaces

- To instruct MCU what and how to control you need to program it.
- MCU was ordinary programmed only in assembly languages.
- But nowadays MCU can be programmed in high-level programming language like C.

Programming MCU

The main function of the MCU is to control things through I/O interfaces

- To instruct MCU what and how to control you need to program it.
- MCU was ordinary programmed only in assembly languages.
- But nowadays MCU can be programmed in high-level programming language like C.

Programming MCU

The main function of the MCU is to control things through I/O interfaces

- To instruct MCU what and how to control you need to program it.
- MCU was ordinary programmed only in assembly languages.
- But nowadays MCU can be programmed in high-level programming language like C.

Programming MCU

To programme MCU you need a particular hardware known as **MCU programmer**.

- Challenge of using programmer;
 - The need for special hardware that is somehow costly.
 - It is difficult to program
- These challenges can be addressed by a **Bootloader**.

Programming MCU

To programme MCU you need a particular hardware known as MCU programmer.

- Challenge of using programmer;
 - The need for special hardware that is somehow costly.
 - It is difficult to program
- These challenges can be addressed by a Bootloader.

Programming MCU

To programme MCU you need a particular hardware known as MCU programmer.

- Challenge of using programmer;
 - The need for special hardware that is somehow costly.
 - It is difficult to program
- These challenges can be addressed by a Bootloader.

Programming MCU

To programme MCU you need a particular hardware known as MCU programmer.

- Challenge of using programmer;
 - The need for special hardware that is somehow costly.
 - It is difficult to program
- These challenges can be addressed by a Bootloader.

Bootloder

Bootloader: A small program that has been loaded onto the MCU.

- This program is programmed just once in the program memory of the MCU using programmer.
- After this the MCU can be programmed without a programmer.

Bootloder

Bootloader: A small program that has been loaded onto the MCU.

- This program is programmed just once in the program memory of the MCU using programmer.
- After this the MCU can be programmed without a programmer.

Outline

1. Building Blocks of the Electronics Systems
2. MCU Basics
3. Arduino Platform
4. Prototyping Circuits
5. Pull-Up and Pull-Down Resistors
6. Embedded Devices
7. Designing Your Project

What is Arduino?

Arduino: An open-source physical computing platform based on;

1. A simple microcontroller (AVR) board and
 2. A development environment for writing software for the board.
- The board can be assembled by hand or purchased preassembled.
 - It offers main three things which make programming of the MCU easy.
 1. An open source Bootloader,
 2. Open Schematic boards,
 3. A development environment,

What is Arduino?

Arduino: An open-source physical computing platform based on;

1. A simple microcontroller (AVR) board and
 2. A development environment for writing software for the board.
- The board can be assembled by hand or purchased preassembled.
 - It offers main three things which make programming of the MCU easy.
 1. An open source Bootloader,
 2. Open Schematic boards,
 3. A development environment,

What is Arduino?

Arduino: An open-source physical computing platform based on;

1. A simple microcontroller (AVR) board and
 2. A development environment for writing software for the board.
- The board can be assembled by hand or purchased preassembled.
 - It offers main three things which make programming of the MCU easy.
 1. An open source Bootloader,
 2. Open Schematic boards,
 3. A development environment,

What is Arduino?

Arduino: An open-source physical computing platform based on;

1. A simple microcontroller (AVR) board and
 2. A development environment for writing software for the board.
- The board can be assembled by hand or purchased preassembled.
 - It offers main three things which make programming of the MCU easy.
 1. An open source Bootloader.
 2. Open Schematic boards.
 3. A development environment.

What is Arduino?

Arduino: An open-source physical computing platform based on;

1. A simple microcontroller (AVR) board and
 2. A development environment for writing software for the board.
- The board can be assembled by hand or purchased preassembled.
 - It offers main three things which make programming of the MCU easy.
 1. An open source Bootloader.
 2. Open Schematic boards.
 3. A development environment.

What is Arduino?

Arduino: An open-source physical computing platform based on;

1. A simple microcontroller (AVR) board and
 2. A development environment for writing software for the board.
- The board can be assembled by hand or purchased preassembled.
 - It offers main three things which make programming of the MCU easy.
 1. An open source Bootloader.
 2. Open Schematic boards.
 3. A development environment.

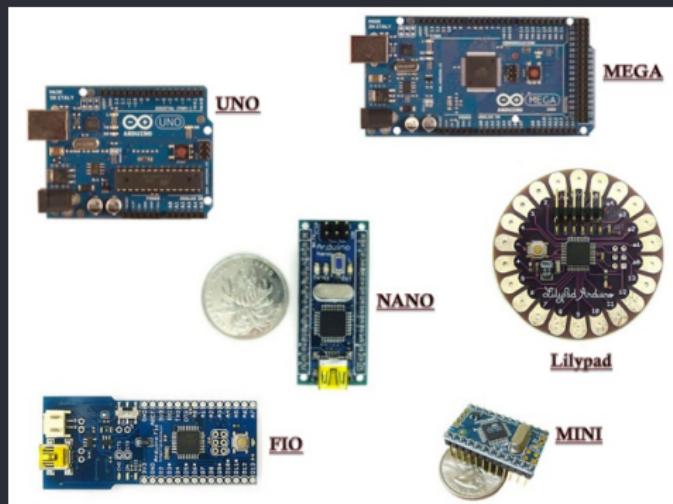
What is Arduino?

Arduino: An open-source physical computing platform based on;

1. A simple microcontroller (AVR) board and
 2. A development environment for writing software for the board.
- The board can be assembled by hand or purchased preassembled.
 - It offers main three things which make programming of the MCU easy.
 1. An open source Bootloader.
 2. Open Schematic boards.
 3. A development environment.

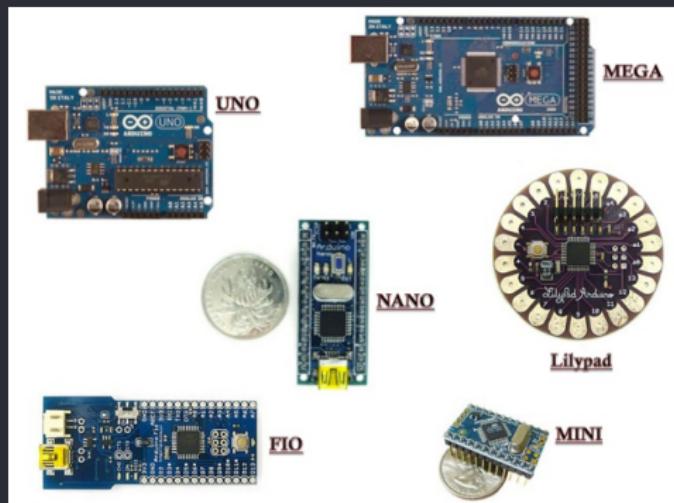
Arduino Hardware

- Uses Atmel Microcontroller (AVR Atmega8 and Atmega168 microcontroller chip).
- Designed to be used with standard C language.
- Exist in several different board variants



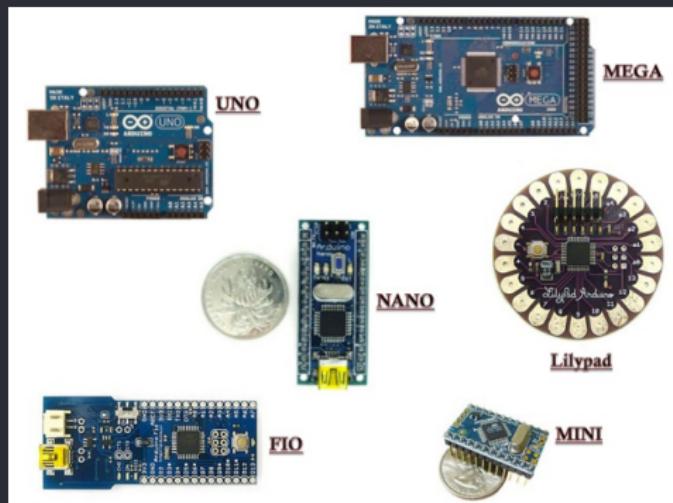
Arduino Hardware

- Uses Atmel Microcontroller (AVR Atmega8 and Atmega168 microcontroller chip).
- Designed to be used with standard C language.
- Exist in several different board variants

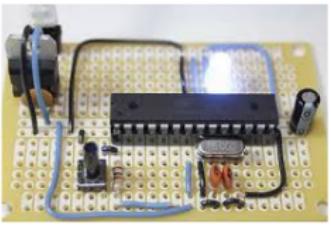
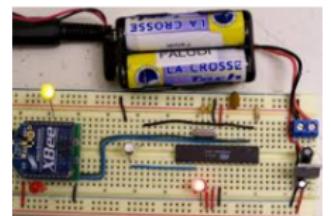
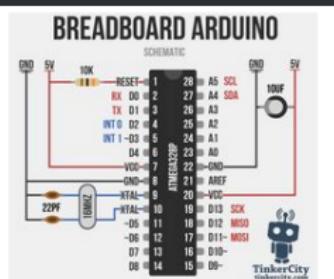
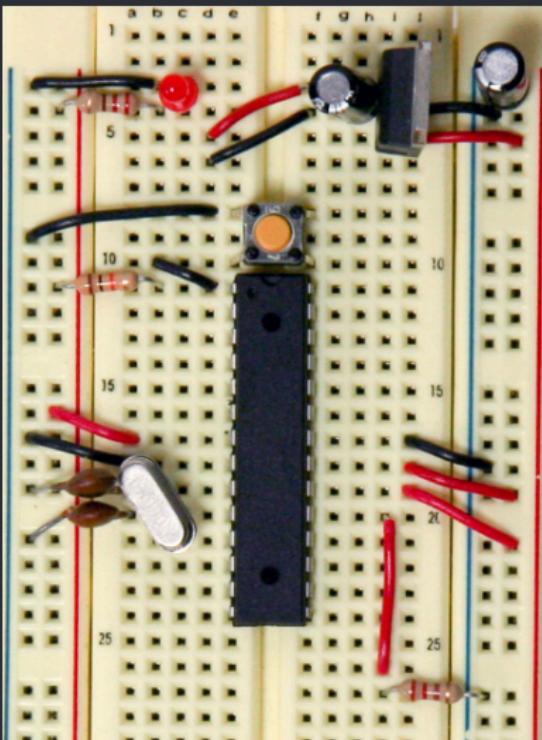


Arduino Hardware

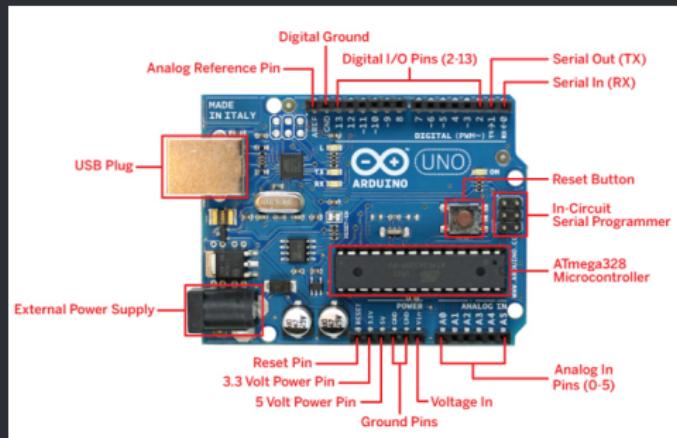
- Uses Atmel Microcontroller (AVR Atmega8 and Atmega168 microcontroller chip).
- Designed to be used with standard C language.
- Exist in several different board variants



Arduino Hardware



Arduino Hardware

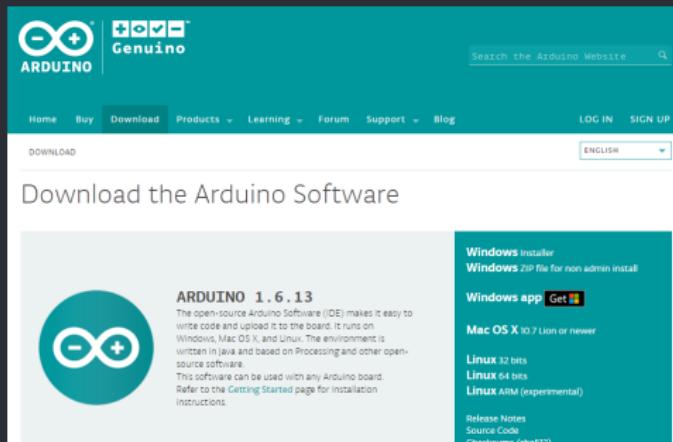


Reset	3v3	5v	Gnd	Vin	Analog In	RX/TX	Digital	PWM(~)	AREF
Resets Arduino sketch on board	3.3 volts in and out	5 volts in and out	Ground	Voltage in for sources over 7V (9V-12V)	Analog inputs, can also be used as Digital	Serial comm. Receive and Transmit	Input or output, HIGH or LOW	Digital pins with output option of PWM	External reference voltage used for analog

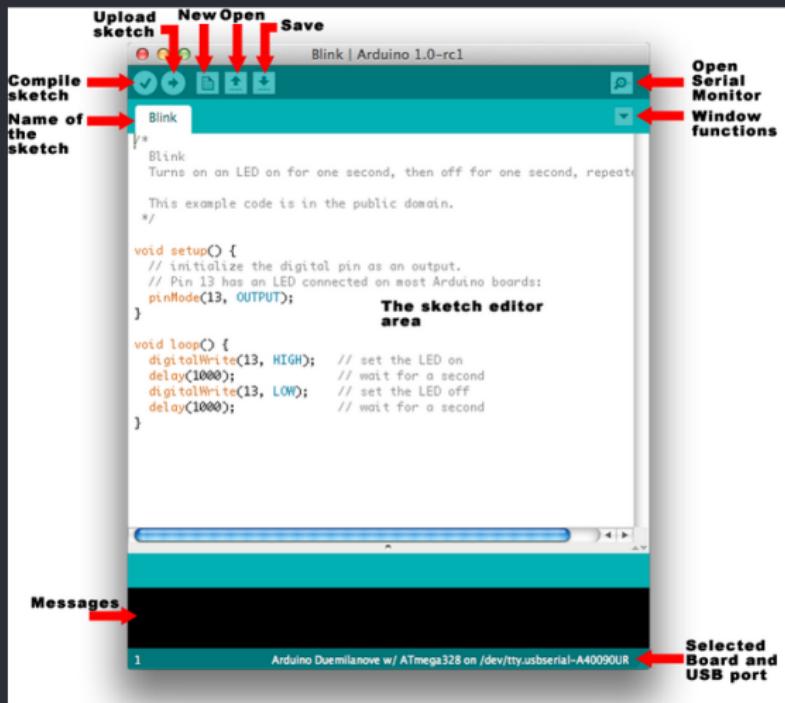
Arduino IDE

A graphical cross platform application written in Java.

- It connects to the Arduino hardware to upload programs and communicate with them.



Arduino IDE



Writing Arduino program

Software written using Arduino are called **sketches**.

- These sketches are written in the text editor (IDE) and saved with the file extension **.ino**.
- A typical sketch consists of two parts or routines.
 - The initialization function `setup()` ⇒ run once at beginning
 - The loop function `loop()` ⇒ run repeatedly, after `setup()`.

Writing Arduino program

Software written using Arduino are called **sketches**.

- These sketches are written in the text editor (IDE) and saved with the file extension **.ino**.
- A typical sketch consists of two parts or routines.
 - The initialization function \Rightarrow `setup()` \Rightarrow run once at beginning
 - The loop function `loop()` \Rightarrow run repeatedly, after `setup()`.

Writing Arduino program

Software written using Arduino are called **sketches**.

- These sketches are written in the text editor (IDE) and saved with the file extension **.ino**.
- A typical sketch consists of two parts or routines.
 - The initialization function \Rightarrow **setup()** \Rightarrow run once at beginning
 - The loop function **loop()** \Rightarrow run repeatedly, after **setup()**.

Writing Arduino program

Software written using Arduino are called **sketches**.

- These sketches are written in the text editor (IDE) and saved with the file extension **.ino**.
- A typical sketch consists of two parts or routines.
 - The initialization function \Rightarrow **setup()** \Rightarrow run once at beginning
 - The loop function **loop()** \Rightarrow run repeatedly, after **setup()**.

Writing Arduino program

Sketch Structure

```
void setup() {  
  
  // put your setup code here, to run once:  
  
}  
  
void loop() {  
  
  // put your main code here, to run repeatedly:  
  
}
```

Example 1

```
// the setup function runs once when you press reset or power the
// board
void setup() {
// initialize digital pin 13 as an output.
pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
digitalWrite(13, HIGH);    // turn the LED on (HIGH is the voltage
// level)
delay(1000);              // wait for a second
digitalWrite(13, LOW);     // turn the LED off by making the
// voltage LOW
delay(1000);              // wait for a second
}
```

Example 2

```
const int buttonPin = 2;      // the number of the pushbutton pin
const int ledPin = 13;        // the number of the LED pin
// variables will change:
int buttonState = 0; // variable for pushbutton status
void setup() {
pinMode(ledPin, OUTPUT); // initialize the LED pin as an output:
pinMode(buttonPin, INPUT); // initialize the pushbutton pin as an
    input:
}
void loop() { // read the state of the pushbutton value:
buttonState = digitalRead(buttonPin); // check if the pushbutton
    is pressed.
if (buttonState == HIGH) { // if it is, the buttonState is HIGH:
    turn LED on:
digitalWrite(ledPin, HIGH);
} else { // turn LED off:
digitalWrite(ledPin, LOW);
}
```

Outline

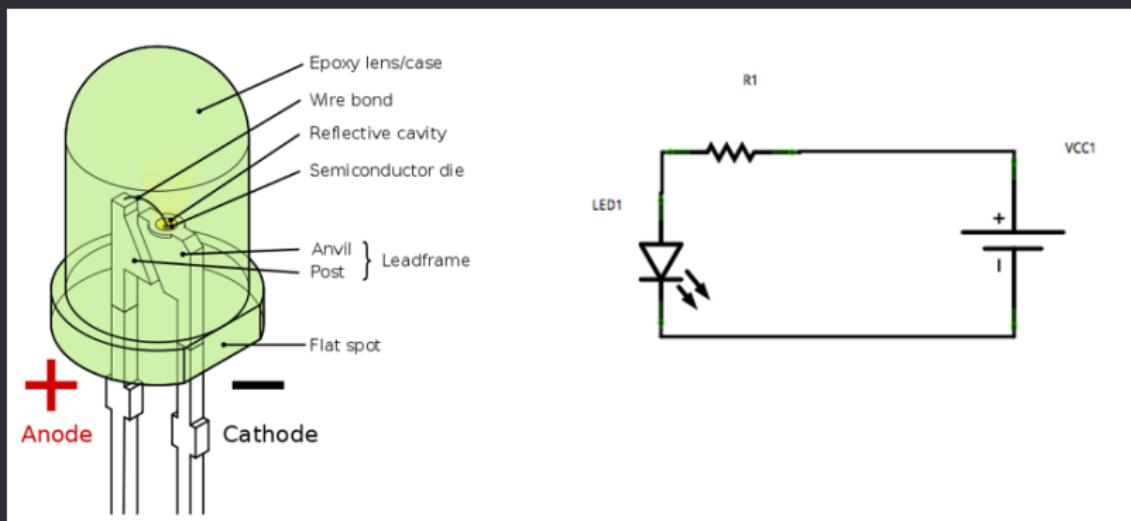
1. Building Blocks of the Electronics Systems
2. MCU Basics
3. Arduino Platform
- 4. Prototyping Circuits**
5. Pull-Up and Pull-Down Resistors
6. Embedded Devices
7. Designing Your Project

Anatomy of Breadboard

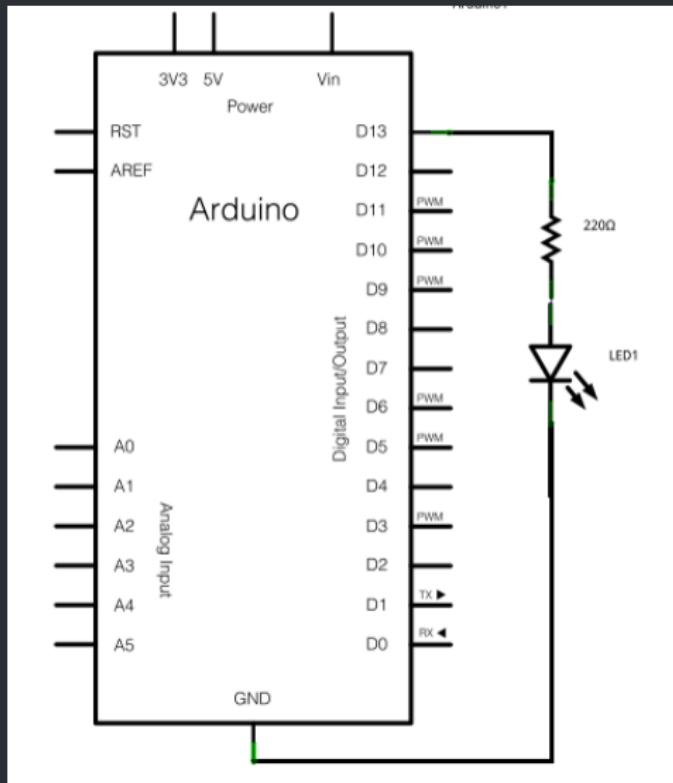
Breadboard: The most useful tools for engineers.



Hello World for a Circuit



Hello World of Arduino



Hello World of Arduino

- Open Arduino IDE and write the code in Example 1
- Connect the Arduino board to the IDE and upload the code.
- Modify the sketch in Example 1 to turns on and off a LED in intervals of 500 Microseconds.
- Modify the sketch in Example 1 to turns on and off 2 LED connected to a digital pin (13 and 12).

Outline

1. Building Blocks of the Electronics Systems
2. MCU Basics
3. Arduino Platform
4. Prototyping Circuits
5. Pull-Up and Pull-Down Resistors
6. Embedded Devices
7. Designing Your Project

Pull-Up and Pull-Down Resistors

Pull-up/down resistors are very common when using MCUs or any digital logic device \Rightarrow to ensure a well-defined logical level at a pin under all conditions

- Consider MCU with one 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 (pulled to VCC) or low (pulled to ground)?
 - It is difficult to tell \Rightarrow This phenomena is referred to as floating.
- To prevent this unknown state:
 - A pull-up or pull-down resistor is used to insure that the pin is in either a high or low state, while also using a low amount of current.

Pull-Up and Pull-Down Resistors

Pull-up/down resistors are very common when using MCUs or any digital logic device \Rightarrow to ensure a well-defined logical level at a pin under all conditions

- Consider MCU with one 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 (pulled to VCC) or low (pulled to ground)?
 - It is difficult to tell \Rightarrow This phenomena is referred to as floating.
- To prevent this unknown state:
 - A pull-up or pull-down resistor is used to insure that the pin is in either a high or low state, while also using a low amount of current.

Pull-Up and Pull-Down Resistors

Pull-up/down resistors are very common when using MCUs or any digital logic device \Rightarrow to ensure a well-defined logical level at a pin under all conditions

- Consider MCU with one 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 (pulled to VCC) or low (pulled to ground)?
 - It is difficult to tell \Rightarrow This phenomena is referred to as floating.
- To prevent this unknown state:
 - A pull-up or pull-down resistor is used to insure that the pin is in either a high or low state, while also using a low amount of current.

Pull-Up and Pull-Down Resistors

Pull-up/down resistors are very common when using MCUs or any digital logic device \Rightarrow to ensure a well-defined logical level at a pin under all conditions

- Consider MCU with one 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 (pulled to VCC) or low (pulled to ground)?
 - It is difficult to tell \Rightarrow This phenomena is referred to as **floating**.
- To prevent this unknown state:
 - A pull-up or pull-down resistor is used to insure that the pin is in either a high or low state, while also using a low amount of current.

Pull-Up and Pull-Down Resistors

Pull-up/down resistors are very common when using MCUs or any digital logic device \Rightarrow to ensure a well-defined logical level at a pin under all conditions

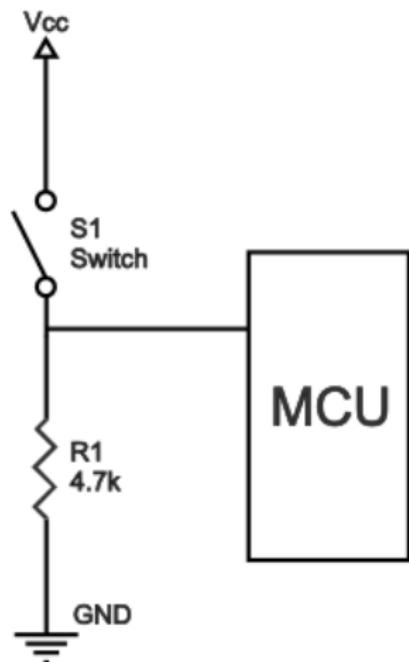
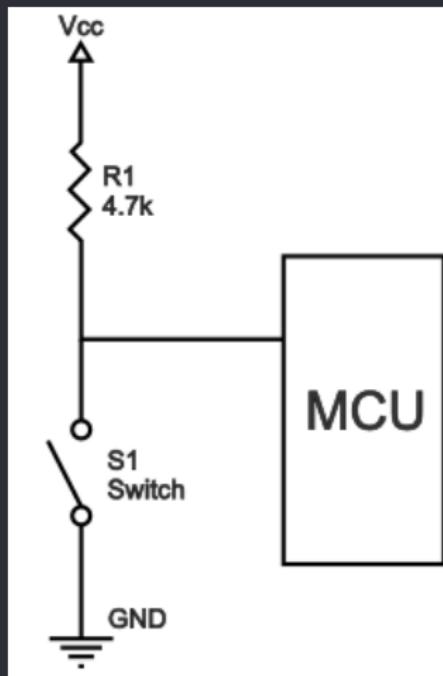
- Consider MCU with one pin configured as an input.
- If there is nothing connected to the pin and your program reads the state of the pin.
 - will it be high (pulled to VCC) or low (pulled to ground)?
 - It is difficult to tell \Rightarrow This phenomena is referred to as **floating**.
- To prevent this unknown state:
 - A pull-up or pull-down resistor is used to insure that the pin is in either a high or low state, while also using a low amount of current.

Pull-Up and Pull-Down Resistors

Pull-up/down resistors are very common when using MCUs or any digital logic device \Rightarrow to ensure a well-defined logical level at a pin under all conditions

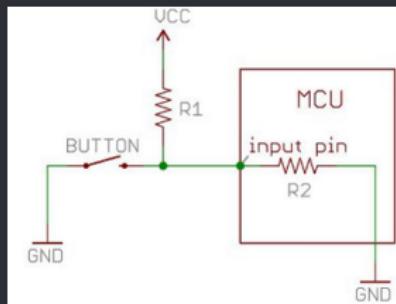
- Consider MCU with one pin configured as an input.
- If there is nothing connected to the pin and your program reads the state of the pin.
 - will it be high (pulled to VCC) or low (pulled to ground)?
 - It is difficult to tell \Rightarrow This phenomena is referred to as **floating**.
- To prevent this unknown state:
 - A pull-up or pull-down resistor is used to insure that the pin is in either a high or low state, while also using a low amount of current.

Pull-Up and Pull-Down Resistors



Pull-Up and Pull-Down Resistors

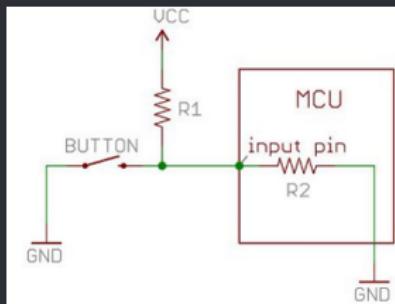
So what value resistor should you choose?. Consider a pull-up resistor below.



1. When the button is pressed, the input pin is pulled low.
 - The value of resistor R1 controls how much current flow from VCC, through the button, and then to ground.
2. When the button is not pressed, the input pin is pulled high.
 - The value of the pull-up resistor controls the voltage on the input pin

Pull-Up and Pull-Down Resistors

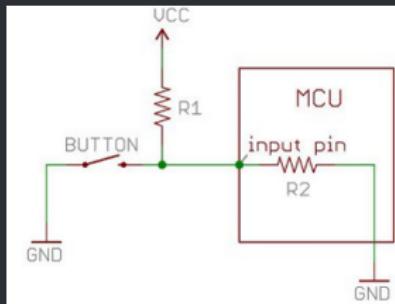
So what value resistor should you choose?. Consider a pull-up resistor below.



1. When the button is pressed, the input pin is pulled low.
 - The value of resistor R1 controls how much current flow from VCC, through the button, and then to ground.
2. When the button is not pressed, the input pin is pulled high.
 - The value of the pull-up resistor controls the voltage on the input pin.

Pull-Up and Pull-Down Resistors

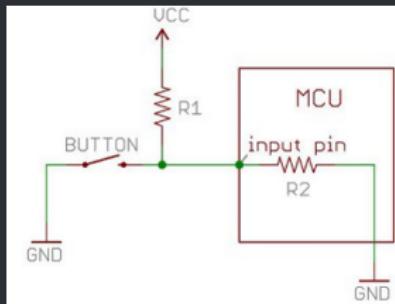
So what value resistor should you choose?. Consider a pull-up resistor below.



1. When the button is pressed, the input pin is pulled low.
 - The value of resistor R1 controls how much current flow from VCC, through the button, and then to ground.
2. When the button is not pressed, the input pin is pulled high
 - The value of the pull-up resistor controls the voltage on the input pin

Pull-Up and Pull-Down Resistors

So what value resistor should you choose?. Consider a pull-up resistor below.



1. When the button is pressed, the input pin is pulled low.
 - The value of resistor R1 controls how much current flow from VCC, through the button, and then to ground.
2. When the button is not pressed, the input pin is pulled high
 - The value of the pull-up resistor controls the voltage on the input pin

Pull-Up and Pull-Down Resistors

So what value resistor should you choose?.

- For condition 1
 - You need a large resistor value ($10\text{ k}\Omega$), but it should not too be large as to conflict with condition 2
- For condition 2:
 - You need to use a pull-up resistor (R_1) that is an order of magnitude ($1/10^{th}$) less than the input impedance (R_2) of the input pin.
 - An input pin on a microcontroller has an impedance that can vary from $10\text{ k}\Omega$ to $10\text{ M}\Omega$.

Pull-Up and Pull-Down Resistors

So what value resistor should you choose?.

- For condition 1
 - You need a large resistor value ($10\text{ k}\Omega$), but it should not too be large as to conflict with condition 2
- For condition 2:
 - You need to use a pull-up resistor (R_1) that is an order of magnitude ($1/10^{10}$) less than the input impedance (R_2) of the input pin.
 - An input pin on a microcontroller has an impedance that can vary from $10\text{ k}\Omega$ to $10\text{ M}\Omega$.

Pull-Up and Pull-Down Resistors

So what value resistor should you choose?.

- For condition 1
 - You need a large resistor value ($10\text{ k}\Omega$), but it should not too be large as to conflict with condition 2
- For condition 2:
 - You need to use a pull-up resistor ($R1$) that is an order of magnitude ($1/10^{th}$) less than the input impedance ($R2$) of the input pin.
 - An input pin on a microcontroller has an impedance that can vary from $10\text{ k}\Omega$ to $10\text{ M}\Omega$.

Pull-Up and Pull-Down Resistors

So what value resistor should you choose?.

- For condition 1
 - You need a large resistor value ($10\text{ k}\Omega$), but it should not too be large as to conflict with condition 2
- For condition 2:
 - You need to use a pull-up resistor ($R1$) that is an order of magnitude ($1/10^{th}$) less than the input impedance ($R2$) of the input pin.
 - An input pin on a microcontroller has an impedance that can vary from $10\text{ k}\Omega$ to $10\text{ M}\Omega$.

Pull-Up and Pull-Down Resistors

So what value resistor should you choose?.

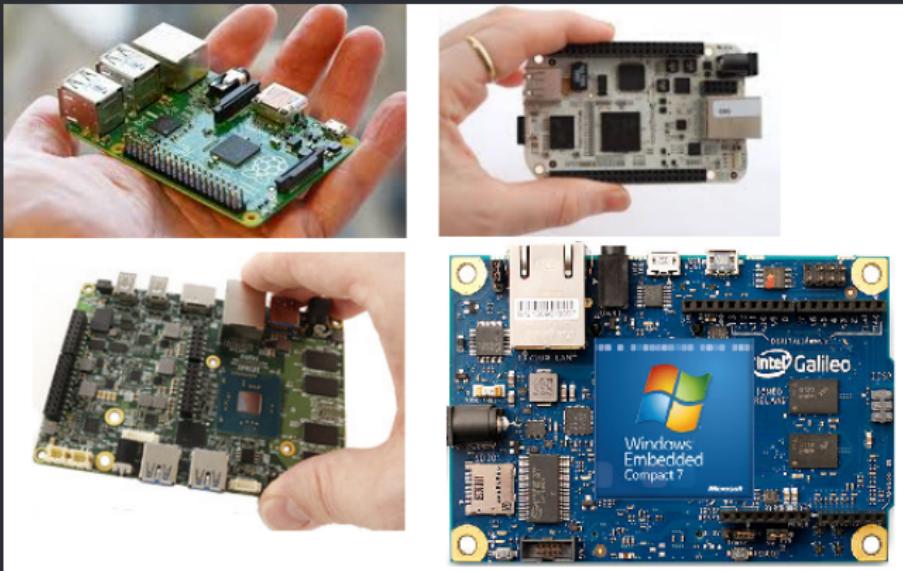
- For condition 1
 - You need a large resistor value ($10\text{ k}\Omega$), but it should not too be large as to conflict with condition 2
- For condition 2:
 - You need to use a pull-up resistor ($R1$) that is an order of magnitude ($1/10^{th}$) less than the input impedance ($R2$) of the input pin.
 - An input pin on a microcontroller has an impedance that can vary from $10\text{ k}\Omega$ to $10\text{ M}\Omega$.

Outline

1. Building Blocks of the Electronics Systems
2. MCU Basics
3. Arduino Platform
4. Prototyping Circuits
5. Pull-Up and Pull-Down Resistors
6. Embedded Devices
7. Designing Your Project

Embedded Devices

Embedded Devices: A general-purpose computer with operating system, and the ability to run multiple programs.



Outline

1. Building Blocks of the Electronics Systems
2. MCU Basics
3. Arduino Platform
4. Prototyping Circuits
5. Pull-Up and Pull-Down Resistors
6. Embedded Devices
7. Designing Your Project

Design Your Project

Design Process:

- Define Your Idea
- Break It Down ⇒ create a block diagram to get an overview of your circuit design.
- A Good Design is Equal to a Good Set of Requirements
- Design each piece of your block diagram (And if you don't know how to – learn it).
- Design your schematics ⇒ . Put the pieces together in one circuit diagram

Design Your Project

Design Process:

- Define Your Idea
- Break It Down ⇒ create a block diagram to get an overview of your circuit design.
- A Good Design is Equal to a Good Set of Requirements
- Design each piece of your block diagram (And if you don't know how to – learn it).
- Design your schematics ⇒ . Put the pieces together in one circuit diagram

Design Your Project

Design Process:

- Define Your Idea
- Break It Down ⇒ create a block diagram to get an overview of your circuit design.
- A Good Design is Equal to a Good Set of Requirements
- Design each piece of your block diagram (And if you don't know how to – learn it).
- Design your schematics ⇒ . Put the pieces together in one circuit diagram

Design Your Project

Design Process:

- Define Your Idea
- Break It Down ⇒ create a block diagram to get an overview of your circuit design.
- A Good Design is Equal to a Good Set of Requirements
- Design each piece of your block diagram (And if you don't know how to – learn it).
- Design your schematics ⇒ . Put the pieces together in one circuit diagram

Design Your Project

Design Process:

- Define Your Idea
- Break It Down ⇒ create a block diagram to get an overview of your circuit design.
- A Good Design is Equal to a Good Set of Requirements
- Design each piece of your block diagram (And if you don't know how to – learn it).
- Design your schematics ⇒ . Put the pieces together in one circuit diagram

Design Your Project

Design Process:

- Simulate your (analog) circuit if needed!.
- Component Selection ⇒ Select Components Wisely (Functionality, Power Consumption, Supporting Circuit Elements etc).
- Test your circuit on bread board.
- Package your project for presentation (Protoboard or PCB Design) ⇒ Requires soldering and desoldering.

Design Your Project

Design Process:

- Simulate your (analog) circuit if needed!.
- Component Selection ⇒ Select Components Wisely (Functionality, Power Consumption, Supporting Circuit Elements etc).
- Test your circuit on bread board.
- Package your project for presentation (Protoboard or PCB Design) ⇒ Requires soldering and desoldering.

Design Your Project

Design Process:

- Simulate your (analog) circuit if needed!.
- Component Selection ⇒ Select Components Wisely (Functionality, Power Consumption, Supporting Circuit Elements etc).
- Test your circuit on bread board.
- Package your project for presentation (Protoboard or PCB Design) ⇒ Requires soldering and desoldering.

Design Your Project

Design Process:

- Simulate your (analog) circuit if needed!.
- Component Selection \Rightarrow Select Components Wisely (Functionality, Power Consumption, Supporting Circuit Elements etc).
- Test your circuit on bread board.
- Package your project for presentation (Protoboard or PCB Design) \Rightarrow Requires soldering and desoldering.

Where to buy equipments

Need Help? ▾ \$ USD

sparkfun SHOP LEARN AVC FORUM DATA

START A PROJECT PRODUCTS BLOG TUTORIALS VIDEOS WISH LISTS DISTRIBUTORS SUPPORT search...

New Products

Top Sellers

SparkFun Originals

2016 Gift Guide

Sale

Gift Certificates

Arduino +

Audio

Books

Breakout Boards

Cables +

Components +

Development Tools +

Dings and Dents

Educators

GPS +

Intel® Edison

HOME / PRODUCT CATEGORIES / SENSORS (207 PRODUCTS)

sort by Most Popular showing 48 per page UPDATE

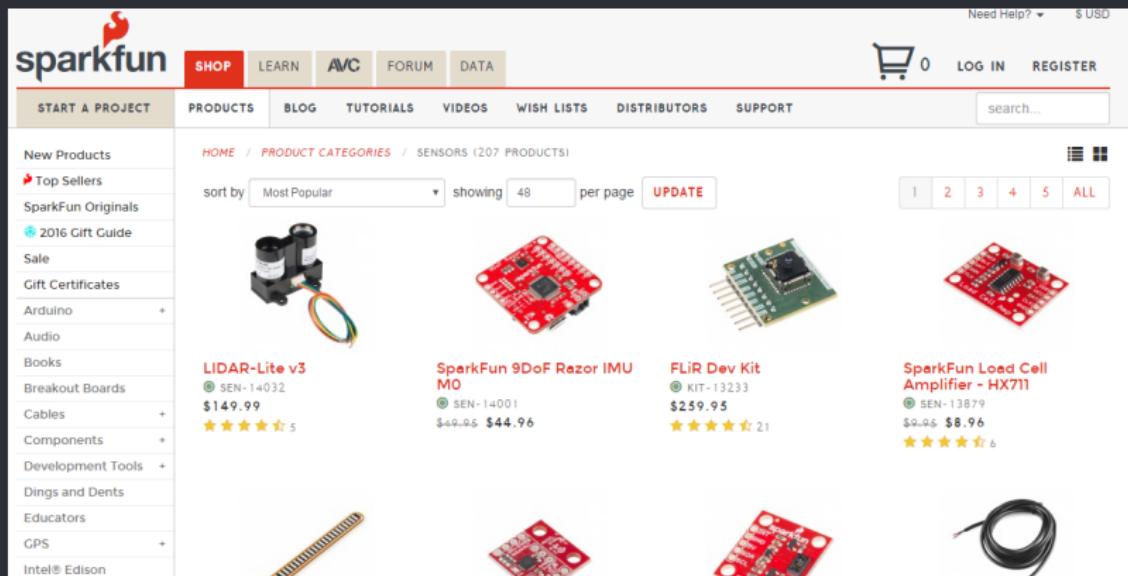
1 2 3 4 5 ALL

LIDAR-Lite v3
@ SEN-14032
\$149.99
★★★★★ 5

SparkFun 9DoF Razor IMU M0
@ SEN-14001
\$49.95 \$44.96

FLIR Dev Kit
@ KIT-13233
\$259.95
★★★★★ 21

SparkFun Load Cell Amplifier - HX711
@ SEN-13879
\$9.95 \$8.96
★★★★★ 6



Where to buy equipments

The screenshot shows the homepage of the Seeed Studio website. At the top, there's a green banner with the text "Free Express Shipping on Orders over \$200!" and a "See Details" button. Below the banner, the Seeed logo is on the left, followed by navigation links: Bazaar, Fusion, Propagate, Community, and Wiki. On the right, there are language (USD), login, sign up, and a shopping cart icon. A search bar is positioned above a main content area. The main content area features a large image of a Quirkbot robotic creature kit, which includes a laptop displaying a software interface and several colorful plastic structural components. A call-to-action bubble says "Bring Your Monster to Live with Quirkbot". The price "\$95" is prominently displayed next to the product name. To the left of the main content, there's a sidebar titled "Category" with a list of products: Arduino, BeagleBone, Raspberry Pi, Linkit, Edison & Galileo, Development Platforms, Wireless, Grove, Interaction, and Tools. A "Grove" badge is visible near the Development Platforms category. At the bottom right of the main content area is a "Feedback & Ideas" link.

Where to buy equipments

Sign In 0 Items 

 [SHOP](#) [BLOG](#) [LEARN](#) [FORUMS](#) [VIDEOS](#)



SENSORS

Sense your way into Open Source Hardware glory with Adafruit's sensor category! Here you can find everything you need to start measuring temperature, motion, force, flow, and more. Check out the [Sensor Pack 900](#) for your beginner sensor needs or the [Soil Temperature/Moisture Sensor](#) for more advanced projects. With a wide and growing range of sensors, Adafruit's Sensors category is the best place for all your needs!

GAS (1)	COIN (2)	LOCATION (GPS) (8)	READERS/SCANNERS (4)
BAROMETRIC PRESSURE (8)	HUMIDITY (12)	MOTION (20)	SOUND/NOISE (9)
BIOMETRIC (5)	LIGHT/COLOR/PHOTO (13)	PROXIMITY (22)	TEMPERATURE (33)
CAMERAS (6)	LIQUID/FLOW (8)	RADIATION / GEIGER (2)	TOUCH (39)



Photo cell (CdS photoresistor)

PRODUCT ID: 161

CdS cells are little light sensors. As the squiggly face is exposed to more light, the resistance goes down. When its light, the resistance is about 5-10KΩ, when dark it goes up to 200KΩ. To use, connect one side of the photo cell (either one, its symmetric) to power (for example 5V) and the other side to your microcontroller's analog input pin. Then connect a 10K pull-down resistor from that analog pin to ground. The voltage on...

Where to buy equipments

Branch Network InfoZone Help

RS Search by keyword, manufacturer part no. or RS part no. Log In | Register 0 £0.00

All Products Our Brands New Products My Account Services

New products
Explore our latest additions, from a Raspberry Pi kit for dummies to pressure calibrators and weighing systems

See latest innovations >

RS Services Quick order Track your order

500,000 Products Available for despatch today

Free next day delivery No minimum order value

Enter your RS order number

Enter your RS account number

Chat Available Sat - Sun Mon-Fri

Where to buy equipments

The screenshot shows the homepage of the Maker Shed website. At the top left is the "Maker Shed" logo. To its right are navigation links: "SHOP", "WHAT'S NEW", "BEST SELLERS", and "SALE". Next are a search icon and a "My Account" link with a shopping cart icon. On the far right are two small thumbnail images and a red "SUBSCRIBE" button. The main visual is a photograph of various electronic components like a Raspberry Pi board, a soldering iron, and breadboards, scattered on a wooden surface. Overlaid on this image is the text "READY. SET. MAKE." in large white letters, followed by "Your homebase for fun, doable projects. We've got something for everyone." Below this is a green button with the text "STAY IN THE KNOW". At the bottom of the page is a white bar containing the text "LET'S EXPLORE...".

Maker Shed

SHOP ▾ WHAT'S NEW ▾ BEST SELLERS SALE

My Account ▾ | Cart

READY. SET. MAKE.

Your homebase for fun, doable projects.
We've got something for everyone.

STAY IN THE KNOW

LET'S EXPLORE...

Where to buy equipments

Featured Brands SUNTRUST

SMARTER STORAGE
Much more storage and security for your files.
On sale Dec 9-11

SHOP NOW

Buyer Protection Help ▾ Save big on our app! Ship to / USD ▾ Language ▾

AliExpress
Smarter Shopping, Better Living!

I'm shopping for... All Categories Cart 0 Wish List Sign In Join My AliExpress

CATEGORIES See All >

- W Women's Clothing
- M Men's Clothing
- P Phones & Accessories
- C Computer & Office
- E Consumer Electronics
- J Jewelry & Watches
- H Home & Garden, Furniture
- B Bags & Shoes
- T Toys, Kids & Baby
- S Sports & Outdoors
- G Health & Beauty
- A Automobiles & Motorcycles

SuperDeals Featured Brands AliExpress Collections Bestselling Tech Discovery Trending Styles

SUPER DEALS
Amazing deals on Blackview smartphones

SHOP NOW

SMARTER STORAGE
Much more storage and security for your files.

Featured Brands On sale

Where to buy equipments

Free shipping to TZ English

PICK YOUR FAVORITE PHONE TO GREET 2017 Choose Your Favorite>> **UP TO 69% OFF**

New Arrivals Bigshot
 Most Popular-Top 50
 Price Range for Your Choice

dealExtreme Multi-language search available 0

Led Bulb, Xiaomi Mi Band 2, Xieomi, DOOGEE, Arduino, Sandisk

ALL DEPARTMENTS

- Consumer Electronics
- PCs / Tablets & Networking
- Electrical & Tools
- Cell Phones & Accessories
- Car Accessories
- Hobbies & Toys
- Lights & Lighting
- Home & Office
- Apparel & Accessories
- Sports & Outdoors
- Health & Beauty

New Arrivals \$0.99 **Top Sellers** **Extreme Deals** **MVP 24 hrs**

TIME TO **GEAR UP YOUR NEW PHONE?**
You've Come to the Right Place

UP TO **83 % OFF**

COMMUNITY

REVIEW & WIN A PRIZE
Check it out >

Household Gadgets
View more >

\$12/1 SALE CARD
Coming

Most Popular Smartphones
View more >

Site to Learn

Make: Gift Guide - What are you getting for the makers in your life?

Metal Sheet & Metal Tube Fiber Laser Cutter



2018 NEW PRODUCT
PLATES AND PIPES
High Speed: 80 mm/min

14 Years Laser Cutter Manufacturer.
Engineers Available Overseas.

森峰[®]
SENFENG

Learn More

ALL PROJECTS

DIFFICULTY  DURATION 

CRAFT & DESIGN
DIGITAL FABRICATION
SCIENCE
TECHNOLOGY
DRONES & VEHICLES
HOME WORKSHOP

Pay what you will for Make: Science + Discovery ebooks! Get 20+ books and support MakerEd. Time is running out! 4d, 19:39:22

CLAIM YOUR BUNDLE

G+ 1k
P 1k
f 632
in 19
S 5
...

Site to Learn

