Lab 7

Introduction to Energia and the CMPE-110 Robot

CMPE-110
Intro to Computer Engineering

R·I·T

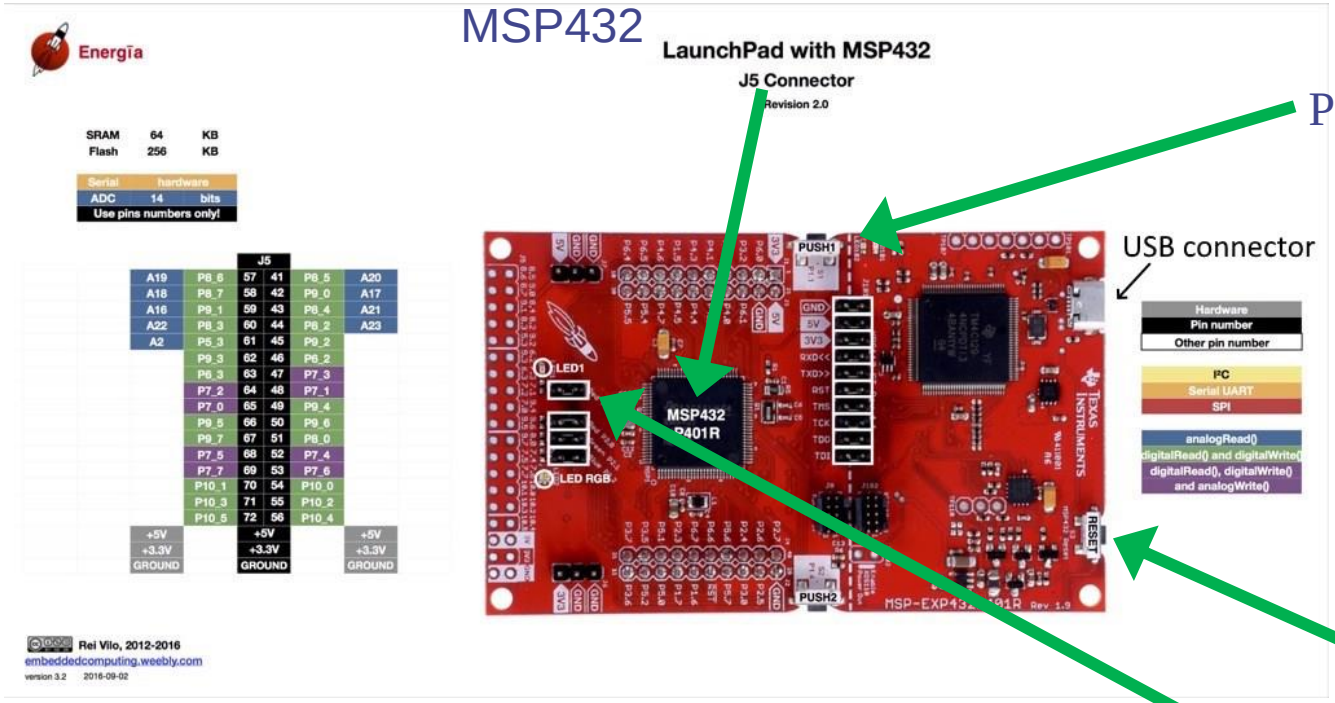
Microcontrollers

- With the release of the Intel 8051 microcontroller almost 30 years ago, a new breed of computer chip was born.
- A microcontroller, as opposed to a microprocessor, is a highly integrated chip which includes everything it needs to be a stand-alone system:
  - CPU
  - Memory
  - Interrupt handlers
  - Timers
  - I/O Ports
- Microcontrollers are hardware oriented and include rich instruction sets to control I/O devices.

ARM Processors

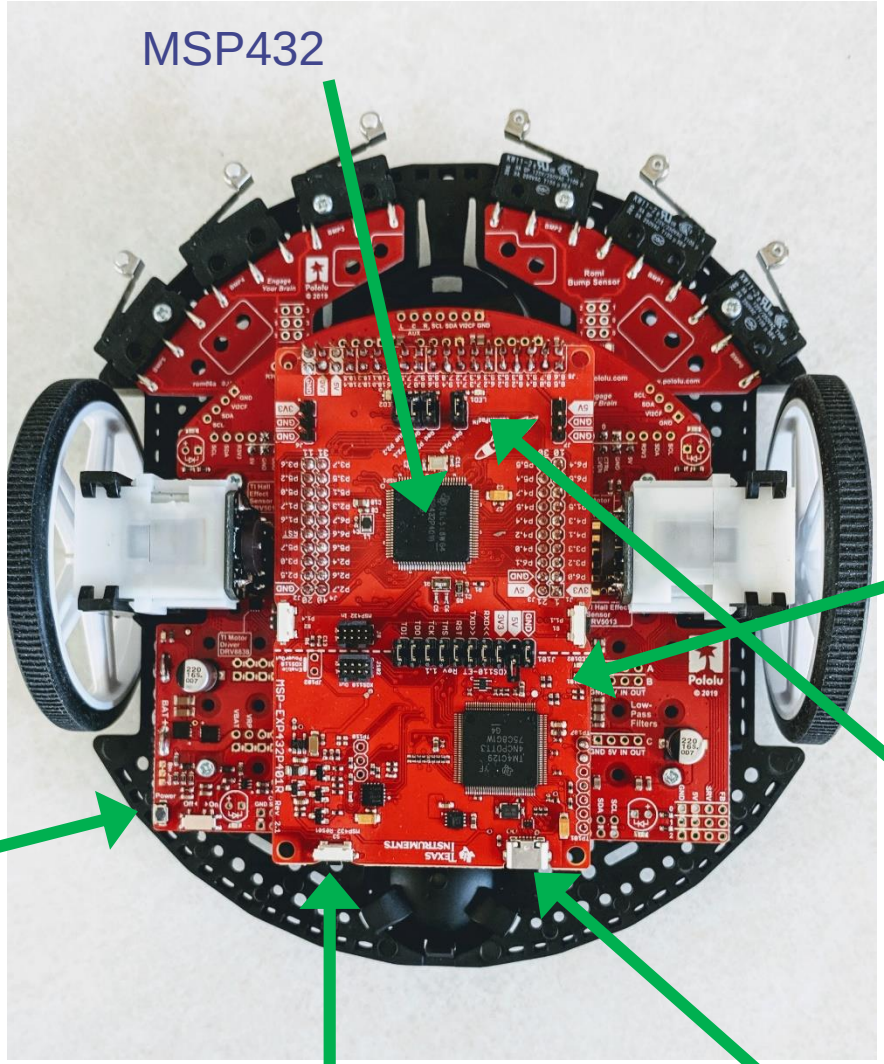
- A family of CPUs based on RISC (reduced instruction set computer) architecture
- Used in devices such as: smartphones, tablets, real-time systems, automobiles, robots, wearables, etc.
- We are using the ARM Cortex-M4 Processor
  - 32-bit microcontroller running at 48 MHz
  - 256 Kbytes of Flash and 64 Kbytes of RAM
  - 48 digital pins programmed to be an input or output

MSP432 Microcontroller



R·I·T

TI-Robot Kit



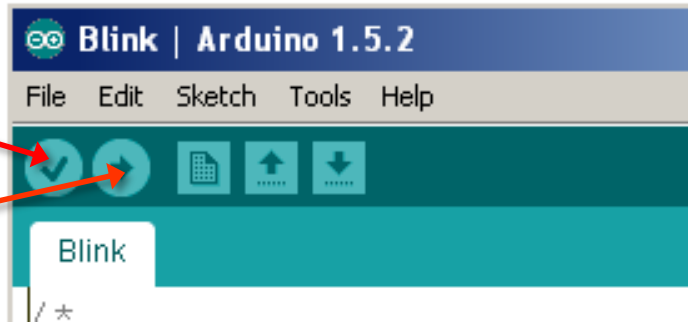
R·I·T

What is Arduino/Energia?

- Arduino is an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board. TI has a special version of Arduino called Energia.
- Arduino Integrated Development Environment (IDE).
  - Programs are called sketches.

Verify/compile

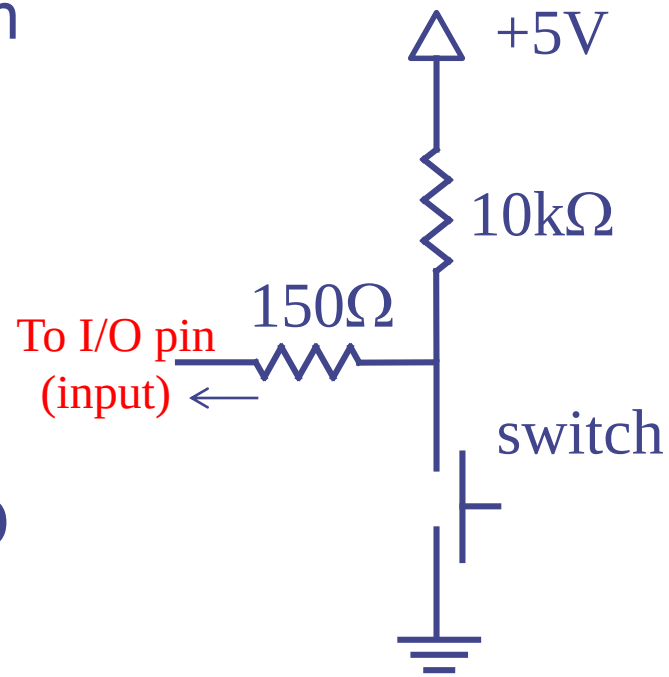
Upload



NOTE: The Energia IDE has a red color

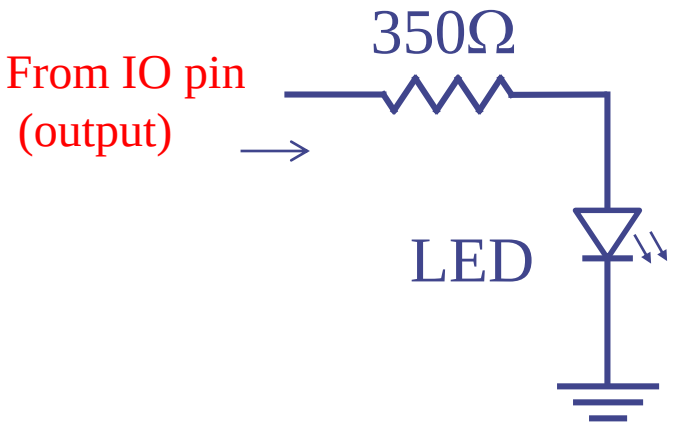
Input Pins

- When you want to take information from the external world (sensors) ***into*** the MCU (e.g. a switch).
- When switch is open, input to I/O pin is only connected to +5V.
  - Read input is 1
- When switch is closed, input to I/O pin is shorted to ground.
  - Read input is 0



Output Pins

- When you want to change the state of something ***out*side** the MCU (e.g. turn a motor or LED on or off).
- Write 1 or HIGH to turn on the LED.
  - Output pin is at +5V
- Write 0 or LOW to turn off the LED.
  - Output pin is at 0V/GND



Arduino Sketch

- Must have
  - setup()

    called once at start of sketch. This is where you do any initialization, such as configuring pin modes and registers.
  - loop()

    Runs the main body of the sketch forver, like while(1) in C/C++.

```
/* Blink  -  turns on an LED for DELAY_ON msec,
then off for DELAY_OFF msec, and repeats

*/
#define LED_PIN  13    // LED on digital pin 13
#define DELAY_ON  1000
#define DELAY_OFF 1000

void setup()
{
  // initialize the digital pin as an output:
  pinMode(LED_PIN, OUTPUT);
}

// loop() method runs forever,
// as long as the Arduino has power

void loop()
{
  digitalWrite(LED_PIN, HIGH);    // set the LED on
  delay(DELAY_ON);  // wait for DELAY_ON msec
  digitalWrite(LED_PIN, LOW);   // set the LED off
  delay(DELAY_OFF);  // wait for DELAY_OFF msec
}
```

R·I·T

Arduino Sketch – Where is Main?

- What happened to the main()???
- The Arduino IDE has is own program file that contains the main() function.

```
int main(void) {
  init();
  setup();
  for (;;)
     loop();
  return(0);
}
```

This initializes Arduino hardware. You don't make your own init() function!

These call the functions in your sketch.

This never gets called as impossible to exit for loop

Arduino Sketch – Set Pins

- Initialize a pin to be input or output
  - pinMode(*pin_no.*, *dir*)

```
pinMode(25, OUTPUT)// pin 25 as output
```

  - Each pin is independent of all others
  - Do this for all pins you are using in a program
- Use output pin to turn LED on and off
  - digitalWrite(*pin_no.*, *value*)

```
digitalWrite(25,HIGH);// LED on
digitalWrite(25,LOW);// LED off
```

R·I·T