Principles and Applications of Microcontrollers

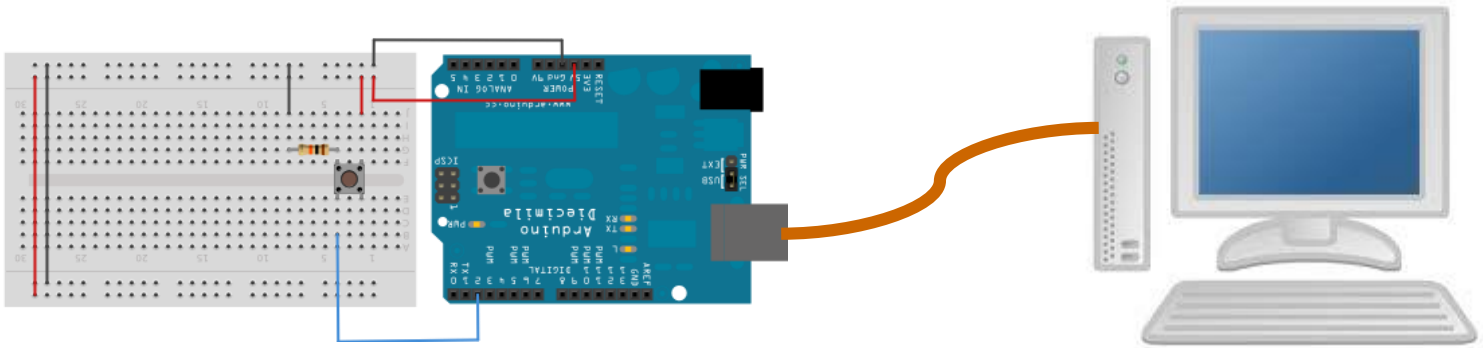
Yan-Fu Kuo
Dept. of Biomechatronics Engineering
National Taiwan University

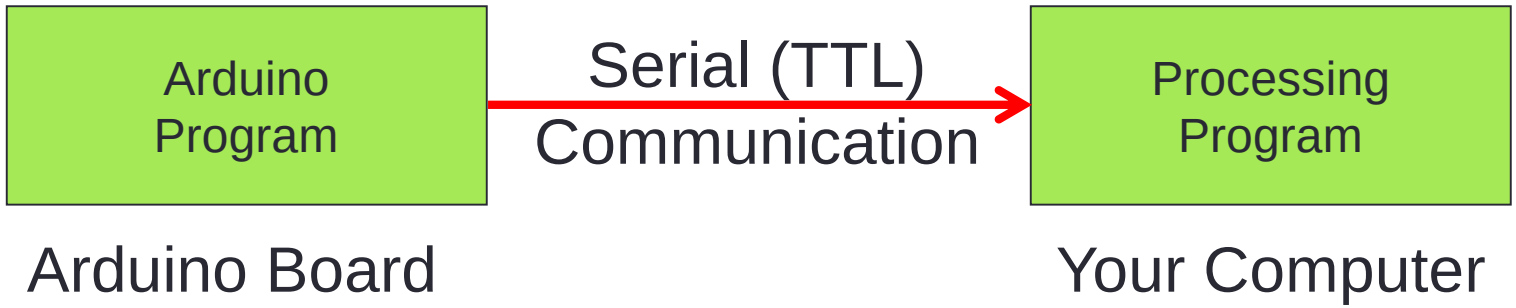
Today:

- Serial communication
- Advanced Arduino programming

Example – Button State Change

- Count how many times the button changes state from “off” to “on”
- Show the counts on the PC screen





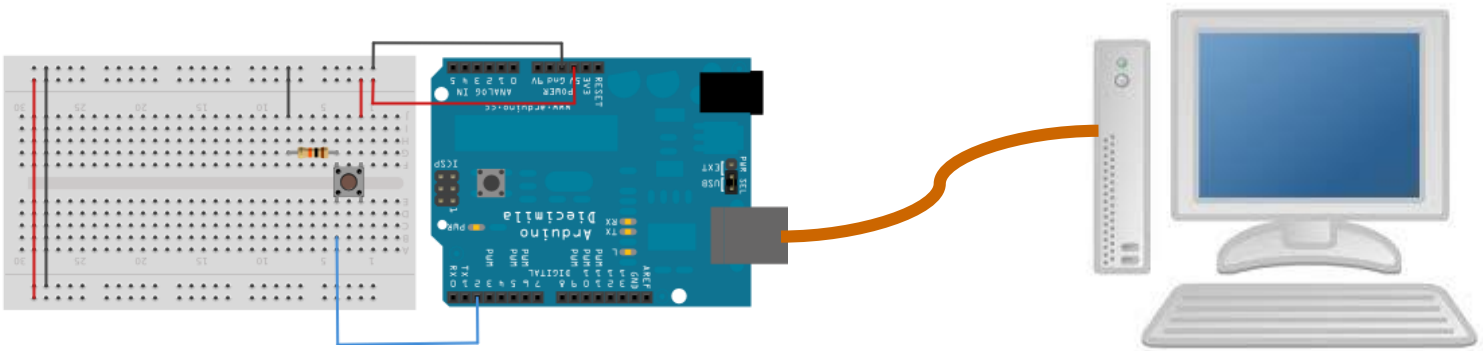
Sketch Code – Button State Change

Link

```
const int buttonPin = 2;                    // the pin that the pushbutton is attached to
int buttonPushCounter = 0;                  // counter for the number of button presses
int buttonState = 0;                        // current state of the button
int lastButtonState = 0;                    // previous state of the button
void setup() {
  pinMode(buttonPin, INPUT);                // initialize the button pin as a input
  Serial.begin(9600);                       // initialize serial communication
}
void loop() {
  buttonState = digitalRead(buttonPin);     // read the pushbutton input pin
  if (buttonState != lastButtonState) {
    if (buttonState == HIGH) {
      buttonPushCounter++;
      Serial.println("on");
      Serial.print("number of button pushes:  ");
      Serial.println(buttonPushCounter);
    }
    else {
      Serial.println("off");
    }
  }
  lastButtonState = buttonState;
}
```

Example – Button Debounce

- Button counter increases spuriously when pressed
- This is because the mechanical limits of the button



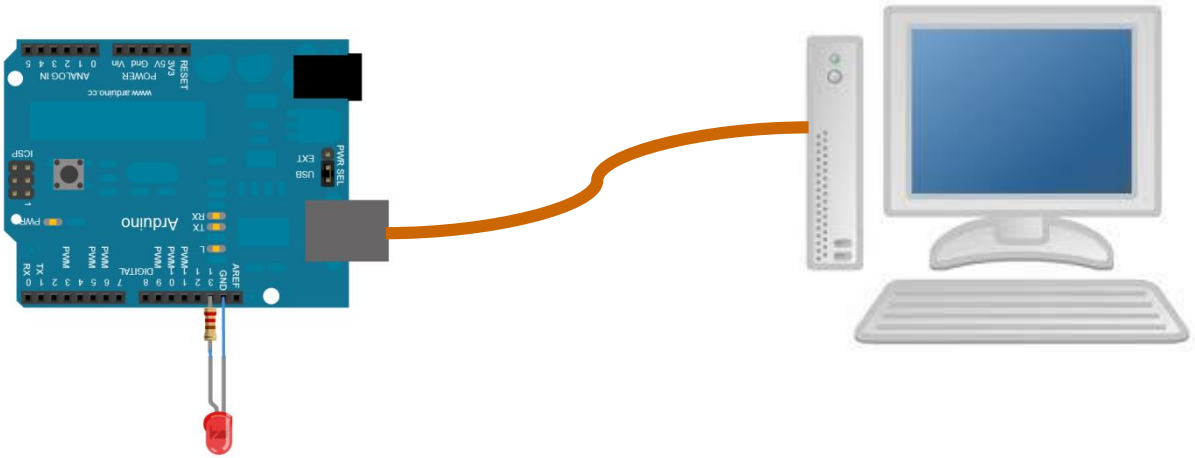
Sketch Code – Button Debounce

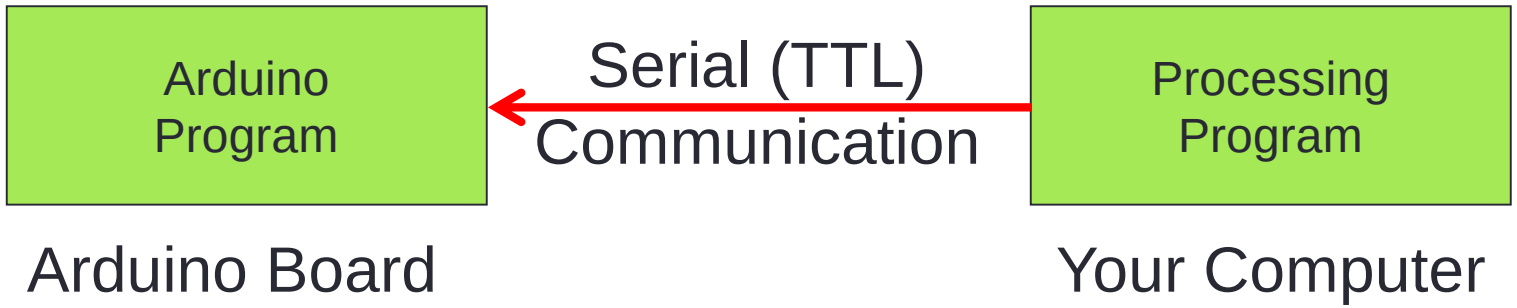
Link

```
const int buttonPin = 2;                    // the number of the pushbutton pin
int buttonState;                            // the current reading from the input pin
int lastButtonState = LOW;                  // the previous reading from the input pin
int buttonPushCounter = 0;                  // counter for the number of button presses
long lastDebounceTime = 0;                  // the last time the output pin was toggled
long debounceDelay = 50;                    // the debounce time
void setup() {
  pinMode(buttonPin, INPUT);
  Serial.begin(9600);
}
void loop() {
  int reading = digitalRead(buttonPin);     // read the state of the switch
  if (reading != lastButtonState) {         // If the switch changed
    lastDebounceTime = millis();            // reset the debouncing timer
  }
  if ((millis() - lastDebounceTime) > debounceDelay) {
    if (reading != buttonState) {           // if the button state has changed
      buttonState = reading;
      if (buttonState == HIGH) {
        buttonPushCounter++;
        Serial.print("number of button pushes: ");
        Serial.println(buttonPushCounter);
      }
    }
  }
  lastButtonState = reading;
}
```

Example – Read Computer Keyboard

- Let Arduino receive signal from the PC
- Press keyboard to turn on or off the LED on Arduino



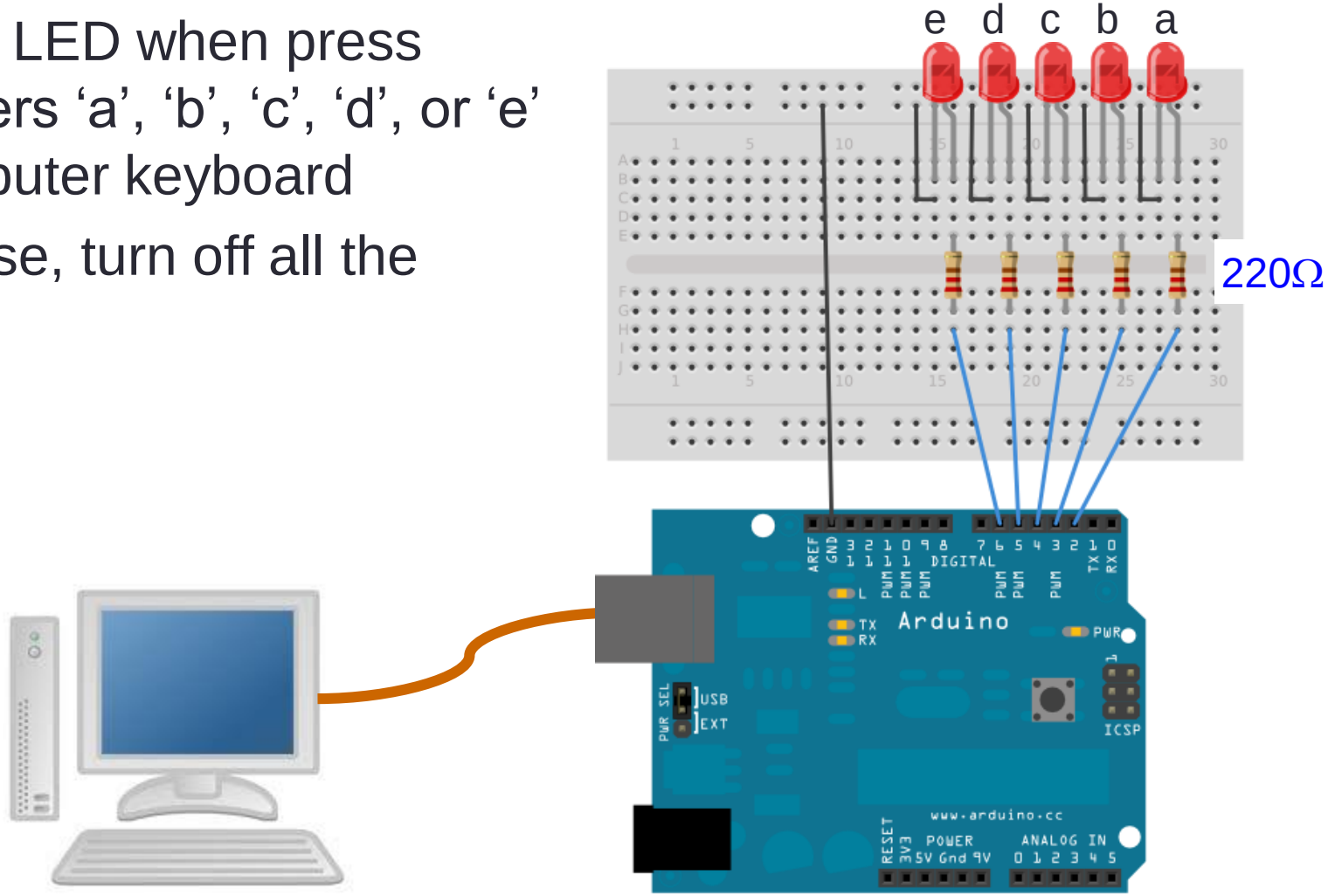


Sketch Code – Read Computer Keyboard

```
int led = 13; // pin 13 has an LED connected on most Arduino boards
char ch;
void setup() {
  pinMode(led, OUTPUT);    // initialize the digital pin as an output
  Serial.begin(9600);      // initialize serial communication at 9600 bps
  Serial.println("+/ - to turn on/off the LED");
}
void loop() {
  if ( Serial.available())
     ch = Serial.read();
  if (ch == '+') {
     Serial.println("On");
     digitalWrite(led, HIGH);
  }
  else if (ch == '-') {
     Serial.println("Off");
     digitalWrite(led, LOW);
  }
  ch = ' ';
}
```

Example – Switch

- Light an LED when press characters ‘a’, ‘b’, ‘c’, ‘d’, or ‘e’ on computer keyboard
- Otherwise, turn off all the LEDs



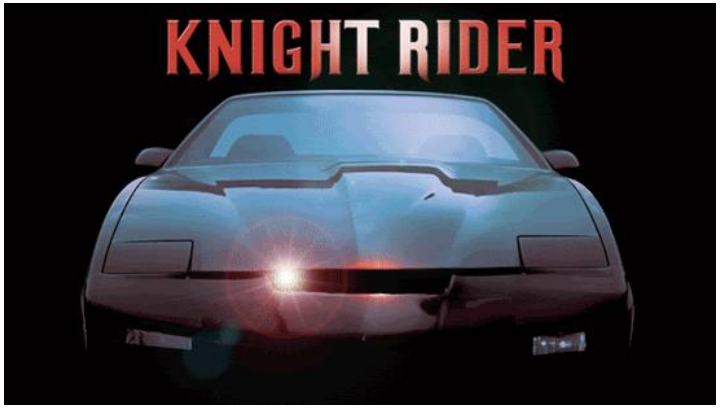
Link

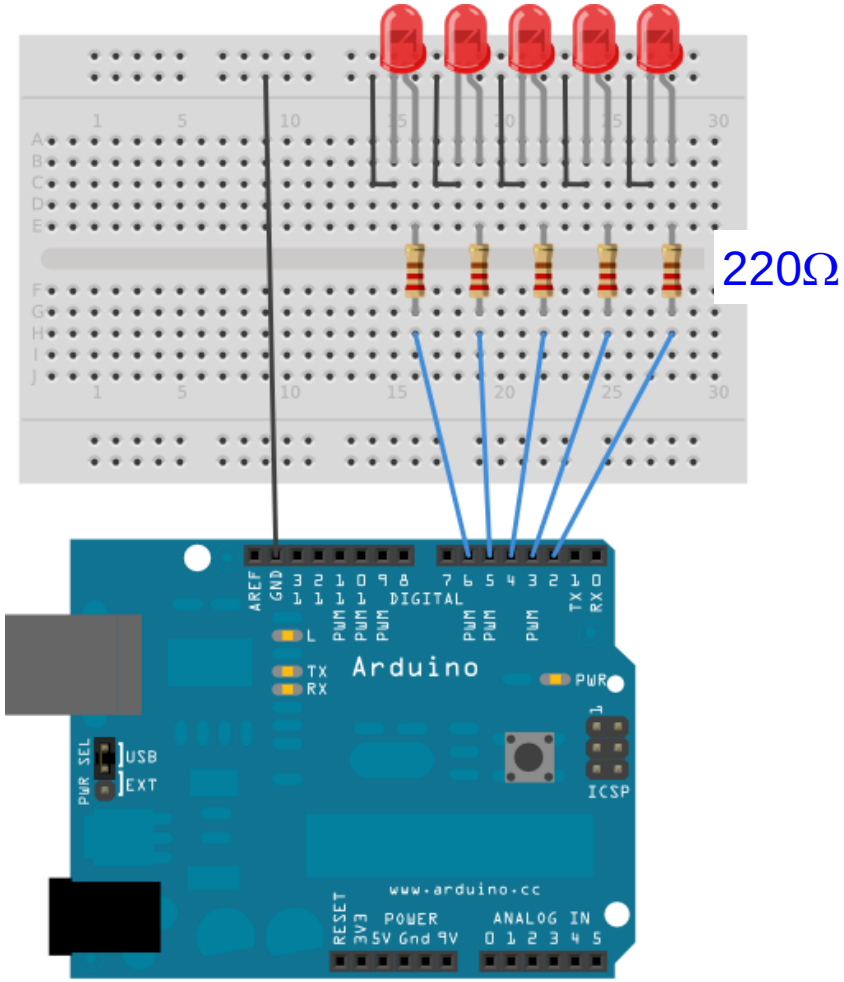
Sketch Code – Switch

```
void setup() {
  Serial.begin(9600);
     // initialize the LED pins
     for (int thisPin = 2; thisPin < 7; thisPin++) {
       pinMode(thisPin,  OUTPUT);
     }
}

void  loop() {
   // read the sensor
  if (Serial.available() > 0) {
    int inByte = Serial.read();

    switch (inByte) {
    case 'a':
      digitalWrite(2, HIGH);
      break;
    case 'b':
      digitalWrite(3, HIGH);
      break;
    case 'c':
      digitalWrite(4, HIGH);
      break;
    case 'd':
      digitalWrite(5, HIGH);
      break;
    case 'e':
      digitalWrite(6, HIGH);
      break;
    default:
      for (int thisPin = 2; thisPin < 7; thisPin++){
        digitalWrite(thisPin, LOW);
      }
    }
  }
}
```

Example – For Loop

- Build a knight rider



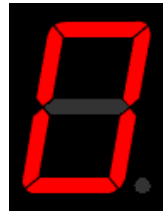


Link

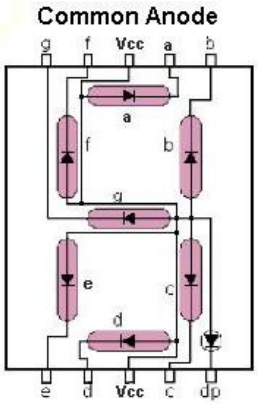
Example – For Loop

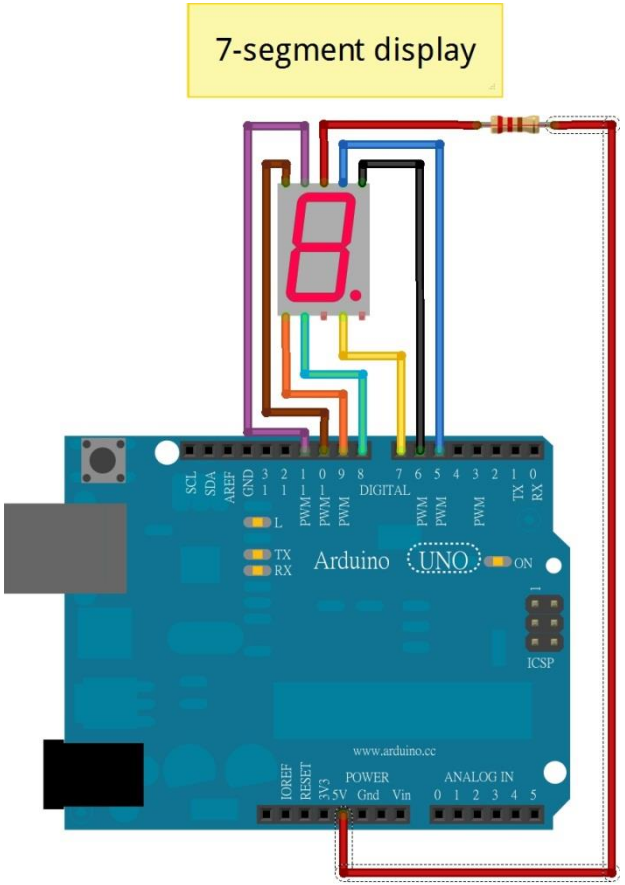
```
int timer = 100;          // delay
void setup() {
  for (int thisPin = 2; thisPin < 7; thisPin++) {    // initialize each pin as an output
    pinMode(thisPin, OUTPUT);
  }
}

void loop() {
  for (int thisPin = 2; thisPin < 7; thisPin++) {  // loop from the lowest pin to the highest
    digitalWrite(thisPin, HIGH);   // turn the pin on:
    delay(timer);
    digitalWrite(thisPin, LOW);   // turn the pin off:
  }
  for (int thisPin = 6; thisPin >= 2; thisPin--) {  // loop from the highest pin to the lowest
    digitalWrite(thisPin, HIGH);   // turn the pin on
    delay(timer);
    digitalWrite(thisPin, LOW);    // turn the pin off
  }
}
```

Example – 7-segment



- Display number 7 on the 7-segment for 3 seconds, then turn it off





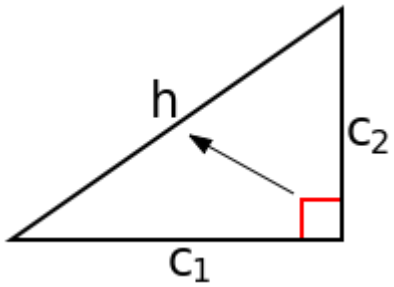
Sketch Code – 7-segment

```
void setup() {
  for(int i=5; i<=11; i++)
    pinMode( i, OUTPUT);
}
void loop() {
    digitalWrite( 5, LOW);
    digitalWrite( 6, LOW);
    digitalWrite( 7, LOW);
    digitalWrite( 8, HIGH);
    digitalWrite( 9, HIGH);
    digitalWrite( 10, HIGH);
    digitalWrite( 11, LOW);
    delay(3000);
    for(int i=5; i<=11; i++)
        digitalWrite( i, HIGH);
}
```

What's wrong with this sketch code?
What should I do to make it work?

Example – Math

- Use Arduino to calculate the hypotenuse of a right-angled triangle, where c1 is equal to 3 and c2 is equal to 4
- Display the numbers on the PC screen



Sketch Code – Math

```
#include "math.h" // include the Math Library
int c1 = 3;
int c2 = 4;
int h;

void setup() {
  Serial.begin(9600);      // initialize serial communication at 9600 bps
  Serial.println("Lets calculate a hypotenuse");
  Serial.print("c1 = ");
  Serial.println(c1);
  Serial.print("c2 = ");
  Serial.println(c2);
  h = sqrt( c1*c1 + c2*c2 );
  Serial.print("h = ");
  Serial.println(h);
}

void loop() {
}
```

Arduino Functions and Libraries

Analog I/O
- analogReference()
- analogRead()
- analogWrite() - *PWM*

Advanced I/O
- tone()
- noTone()
- shiftOut()
- shiftIn()
- pulseIn()

Time
- millis()
- micros()
- delay()
- delayMicroseconds()

Math
- min()
- max()
- abs()
- constrain()
- map()
- pow()
- sqrt()

Trigonometry
- sin()
- cos()
- tan()

Random Numbers
- randomSeed()
- random()

Bits and Bytes
- lowByte()
- highByte()
- bitRead()
- bitWrite()
- bitSet()
- bitClear()
- bit()

Interrupts
- interrupts()
- noInterrupts()

Communication
- Serial
- Stream

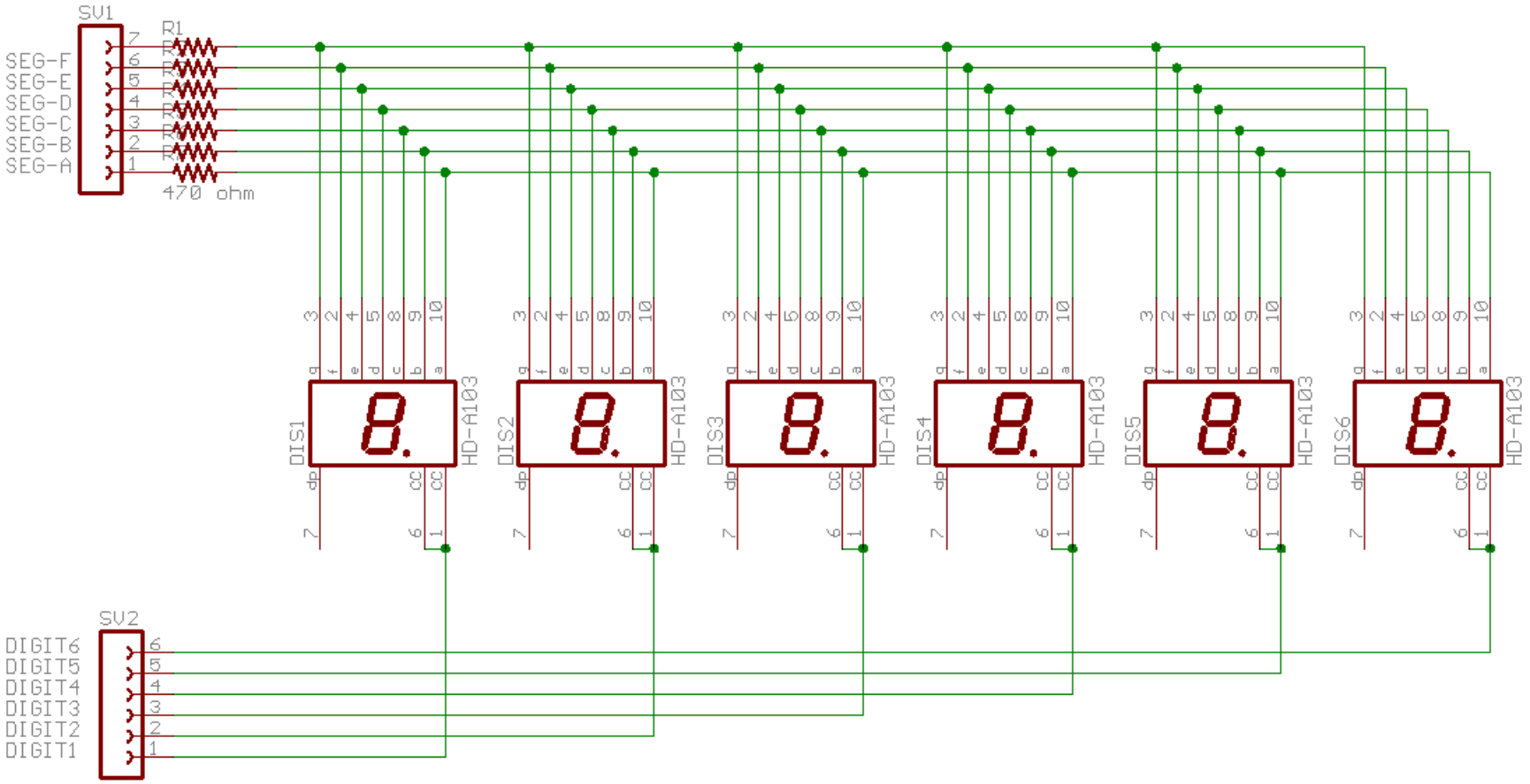
Time-related Functions

- millis():
  - Returns the number of milliseconds since the Arduino board began running the current program
  - Overflow (go back to zero) after approximately 50 days
- micros():
  - Returns the number of microseconds since the Arduino board began running the current program
  - Overflow (go back to zero) after approximately 70 minutes
  - On 16 MHz boards, this function has a resolution of four microseconds

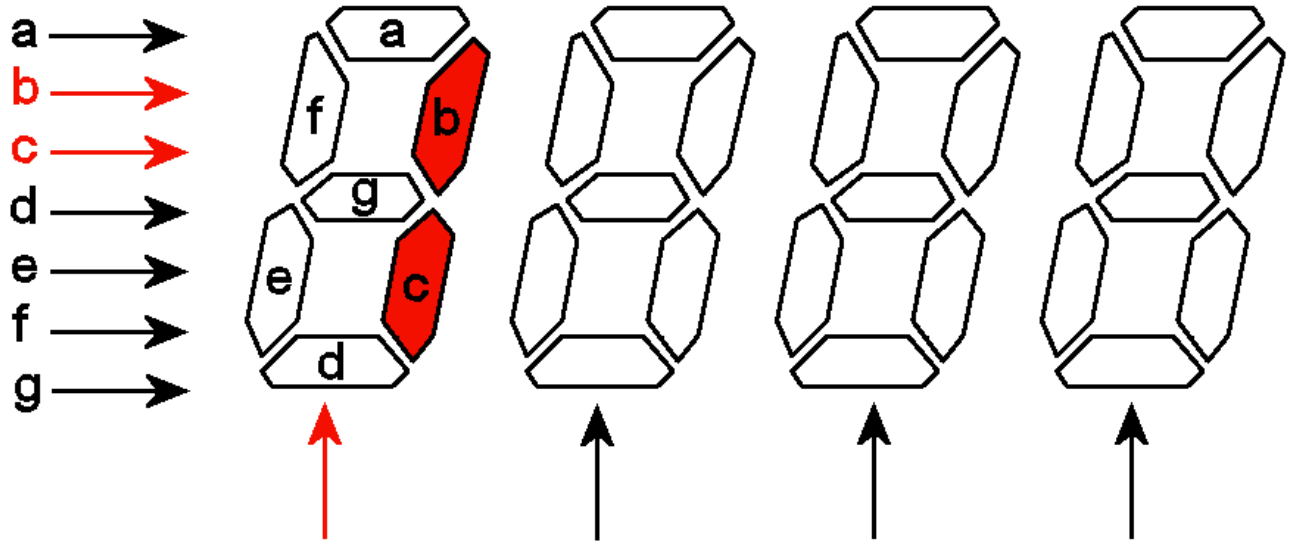
What Have We Learned So Far?

- There is a protocol that Arduino can communicate with a PC – serial
- Structuring program works on Arduino
- There is a component in the microcontroller that counts the time – millis() and mircos()

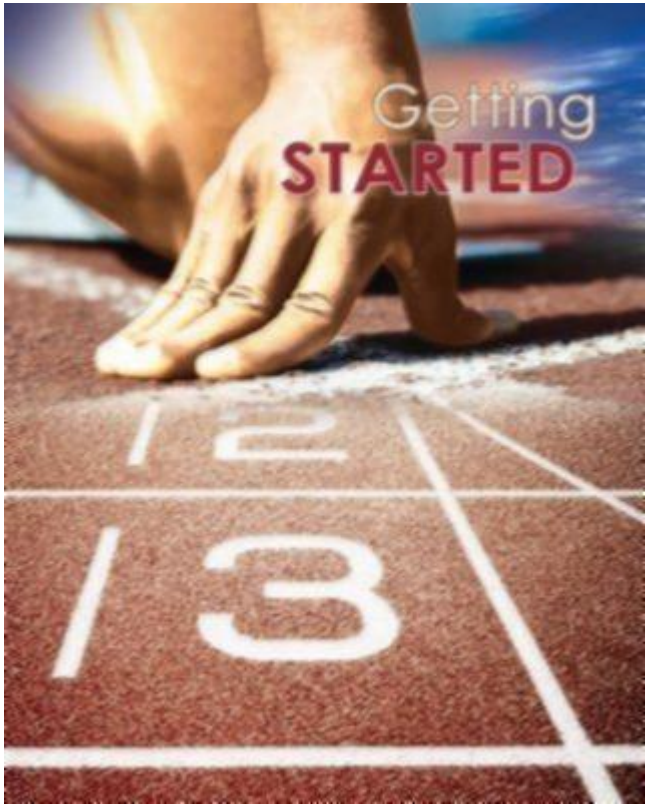
Multiplexed 7-segment LED



Multiplexed 7-segment LED



Getting Started



Reference

- http://www.arduino.cc/
- ATmega328P data sheet