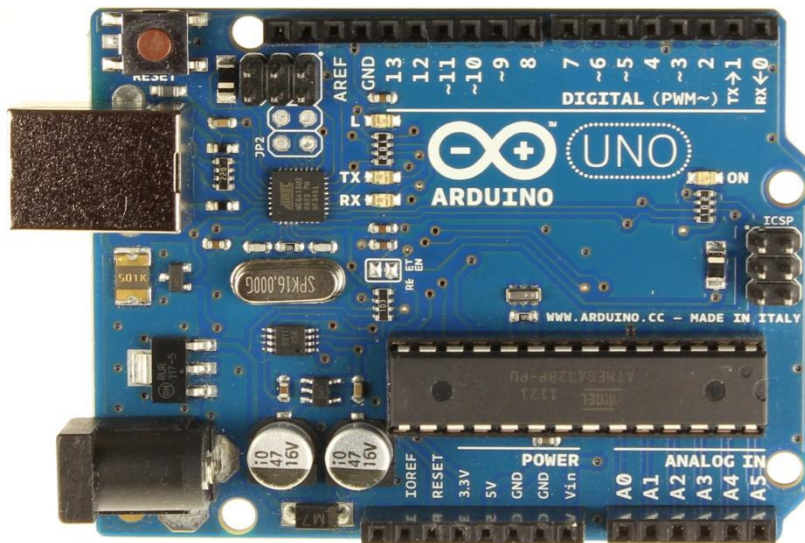


# BoeBots with Arduinos

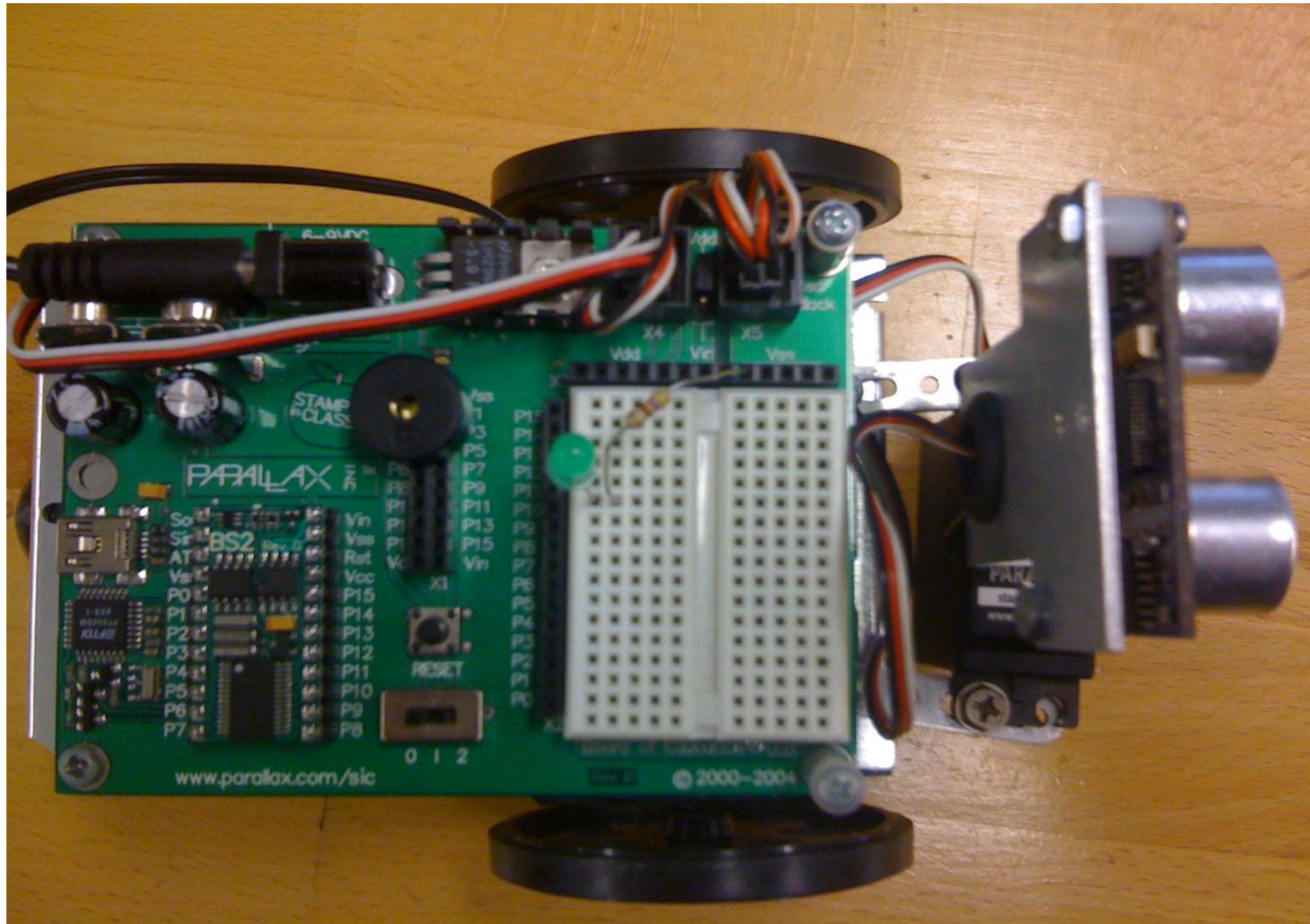


Malika Meghjani

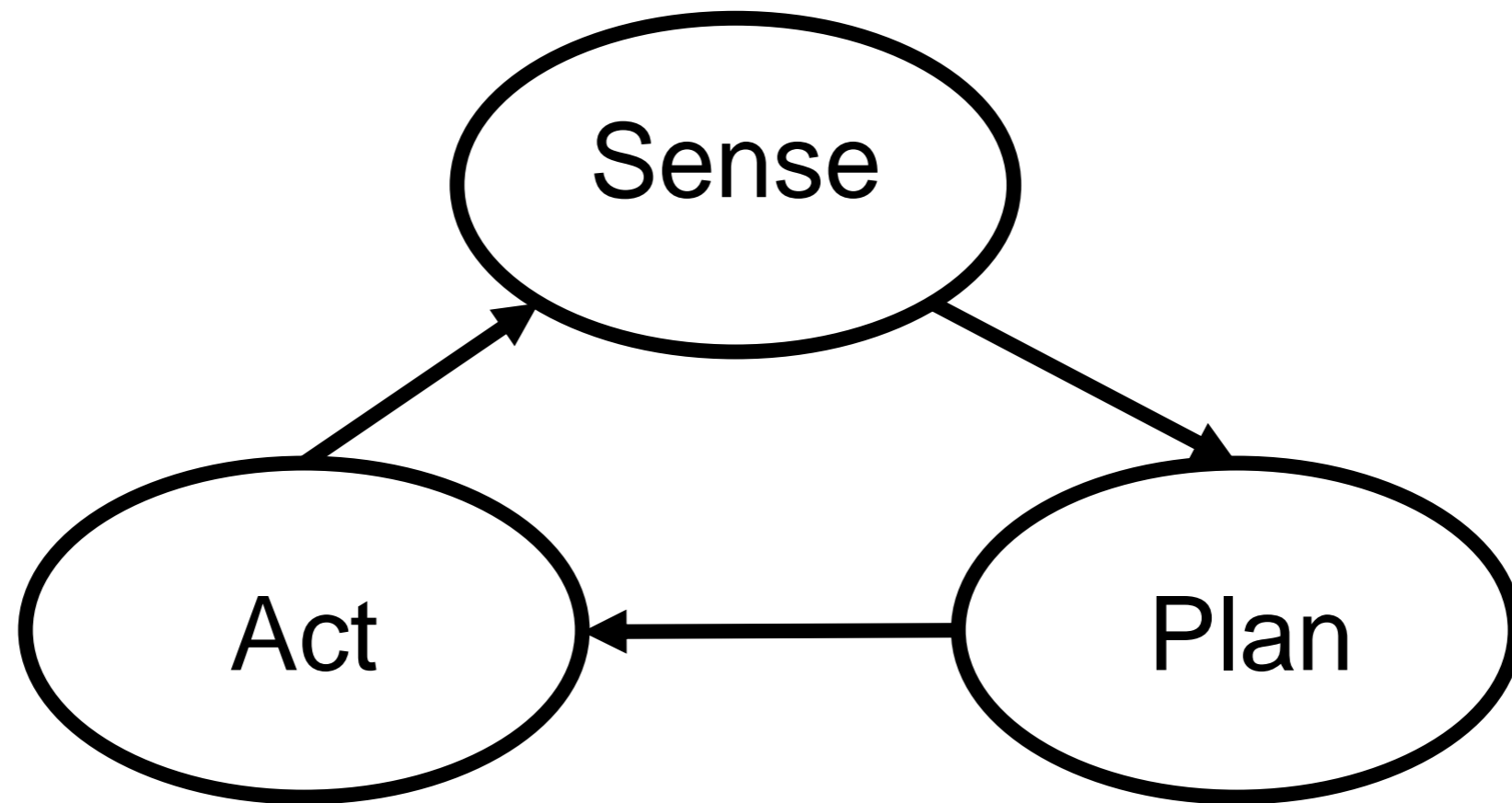
[malika@cim.mcgill.ca](mailto:malika@cim.mcgill.ca)

adapted from boeBot tutorial by Yogesh Girdhar

# Boe-bot



# Boeobot Architecture



# Microcontroller: the brain

- Mini computer on a single chip
- Designed to perform single task
- **Examples:** keyboard, mouse, remote controller, mobile phone, microwave

# Arduino Uno

Microcontroller ATmega328

Operating Voltage 5V

Input Voltage (limits) 6-20V

Digital I/O Pins 14

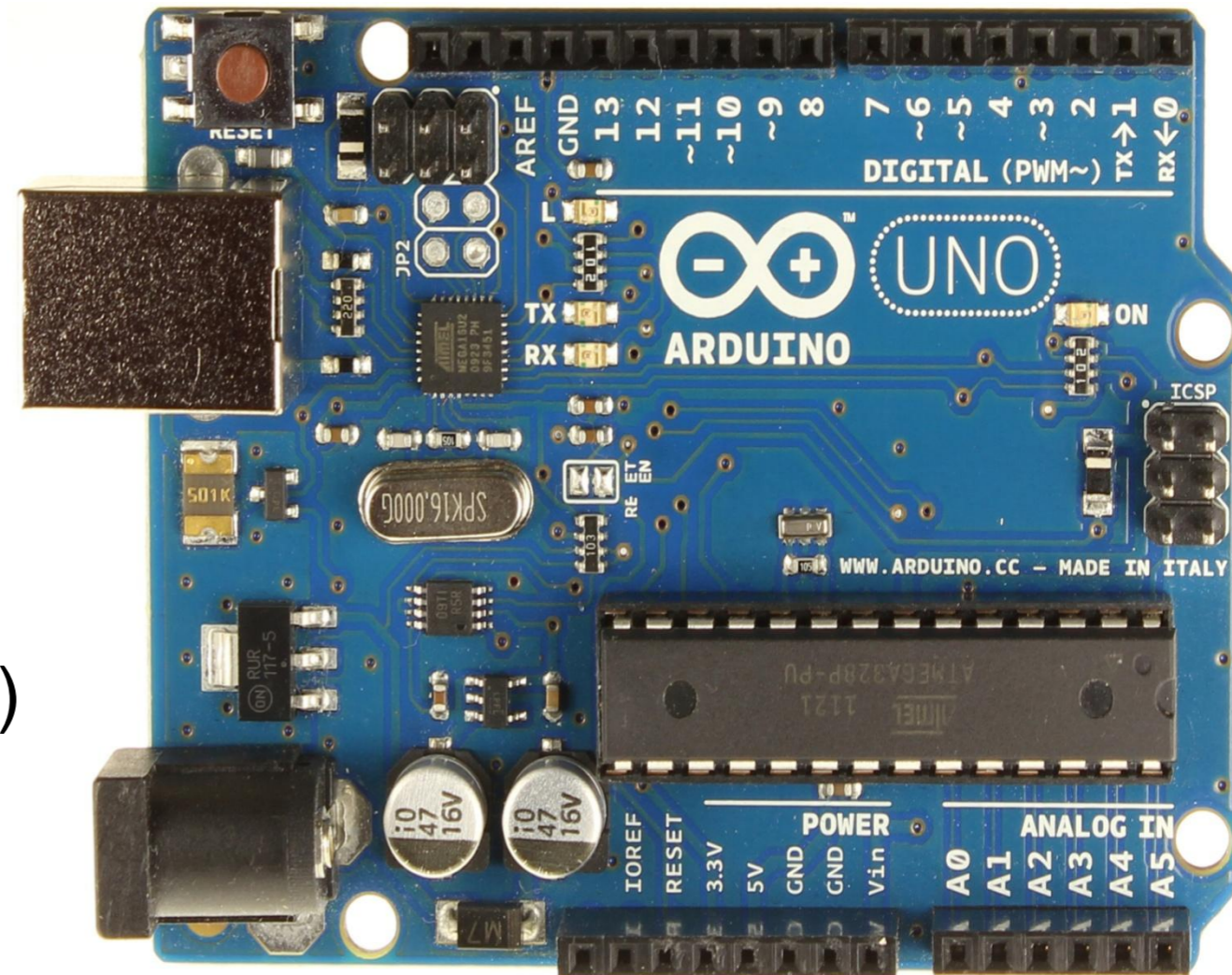
Analog Input Pins 6

Flash Memory 32 KB (ATmega328)

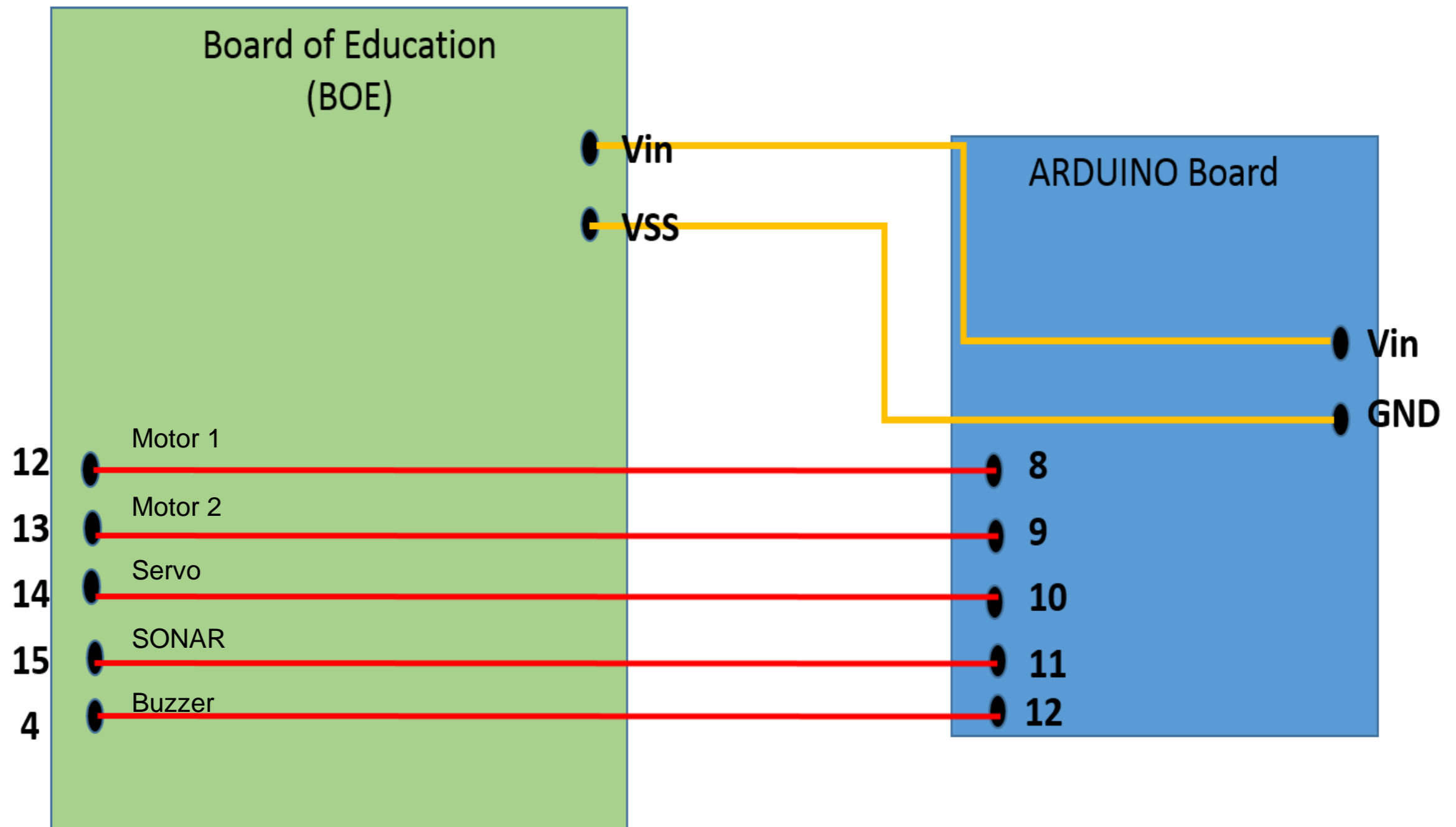
EEPROM 1 KB (ATmega328)

SRAM 2 KB (ATmega328)

Clock Speed 16 MHz



# BoeBot-Arduino Interface



# Software

- Programming in C
- Sketches
- Sketchbook
- <http://arduino.cc/en/Main/Software>

# Getting started

```
void setup() {
```

```
  // put your setup code here, to run once
```

```
}
```

```
void loop() {
```

```
  // put your main code here, to run repeatedly
```

```
}
```

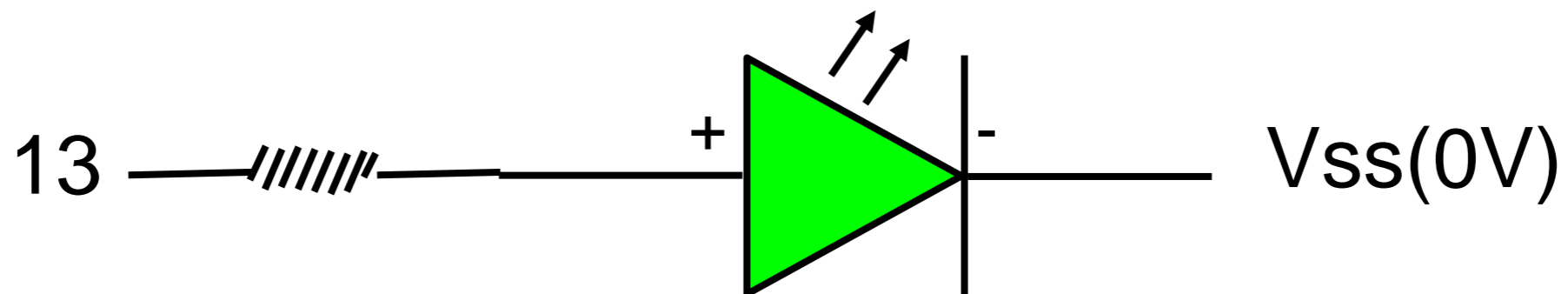


# Hello World

```
void setup() {  
  Serial.begin(9600); // initialize serial comm.  
}  
  
void loop() {  
  Serial.println("Hello World!");  
}
```

# Setting Pins to HIGH/LOW

- *digitalWrite(13, HIGH)*  
*PIN 13 now has +5V*
- *digitalWrite(13, LOW)*  
*PIN 13 now has 0V*



# Blinking LED

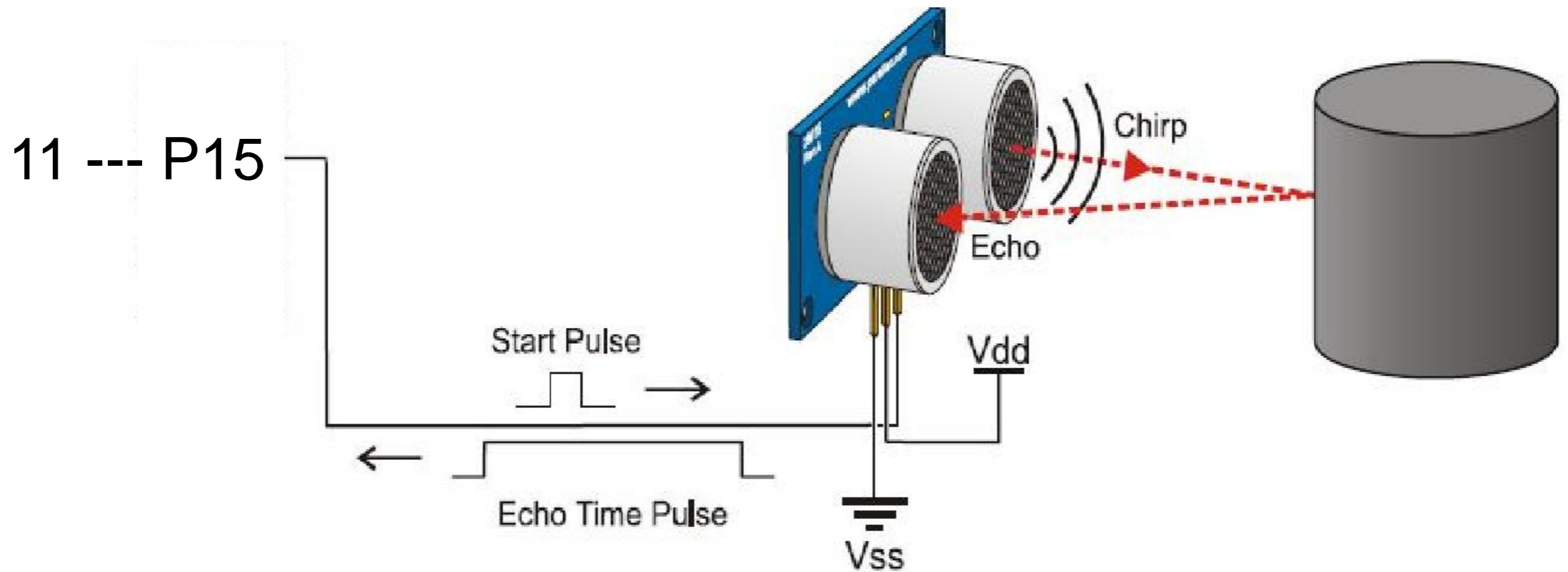
**Refer:** <~/arduino-1.0.5/examples/01.Basics/Blink/Blink.ino>

```
int led = 13; // Pin 13 has an LED connected on most Arduino boards

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

void loop() {
    digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000); // wait for a second
    digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
    delay(1000); // wait for a second
}
```

# Ultrasonic Distance Sensor (SONAR)



# Detecting Distance

**Refer:** ~/arduino-1.0.5/examples/06.Sensors/Ping/Ping.ino

```
const int pingPin = 11;    // Pin 11 is connected to the SONAR of the boeobot

void setup() {

    Serial.begin(9600);    // initialize serial communication

}

void loop() {

    long duration, inches, cm;

    pinMode(pingPin, OUTPUT); // make the pingPin as output to send a pulse

    digitalWrite(pingPin, LOW);

    delayMicroseconds(2);    // send a LOW pulse for 2 microseconds

    digitalWrite(pingPin, HIGH);

    delayMicroseconds(5);    // send a HIGH pulse for 5 microseconds .....
```

# Detecting Distance

.....

```
digitalWrite(pingPin, LOW);

pinMode(pingPin, INPUT);           // use same pin to read signal

duration = pulseIn(pingPin, HIGH);

inches = microsecondsToInches(duration);

cm = microsecondsToCentimeters(duration);

Serial.print(inches);

Serial.print("in, ");

Serial.print(cm);

Serial.print("cm");

Serial.println();

delay(100);

}
```

# Detecting Distance

## Sub routines:

```
long microsecondsToInches(long microseconds)
```

```
{
```

```
    return microseconds / 74 / 2;
```

```
}
```

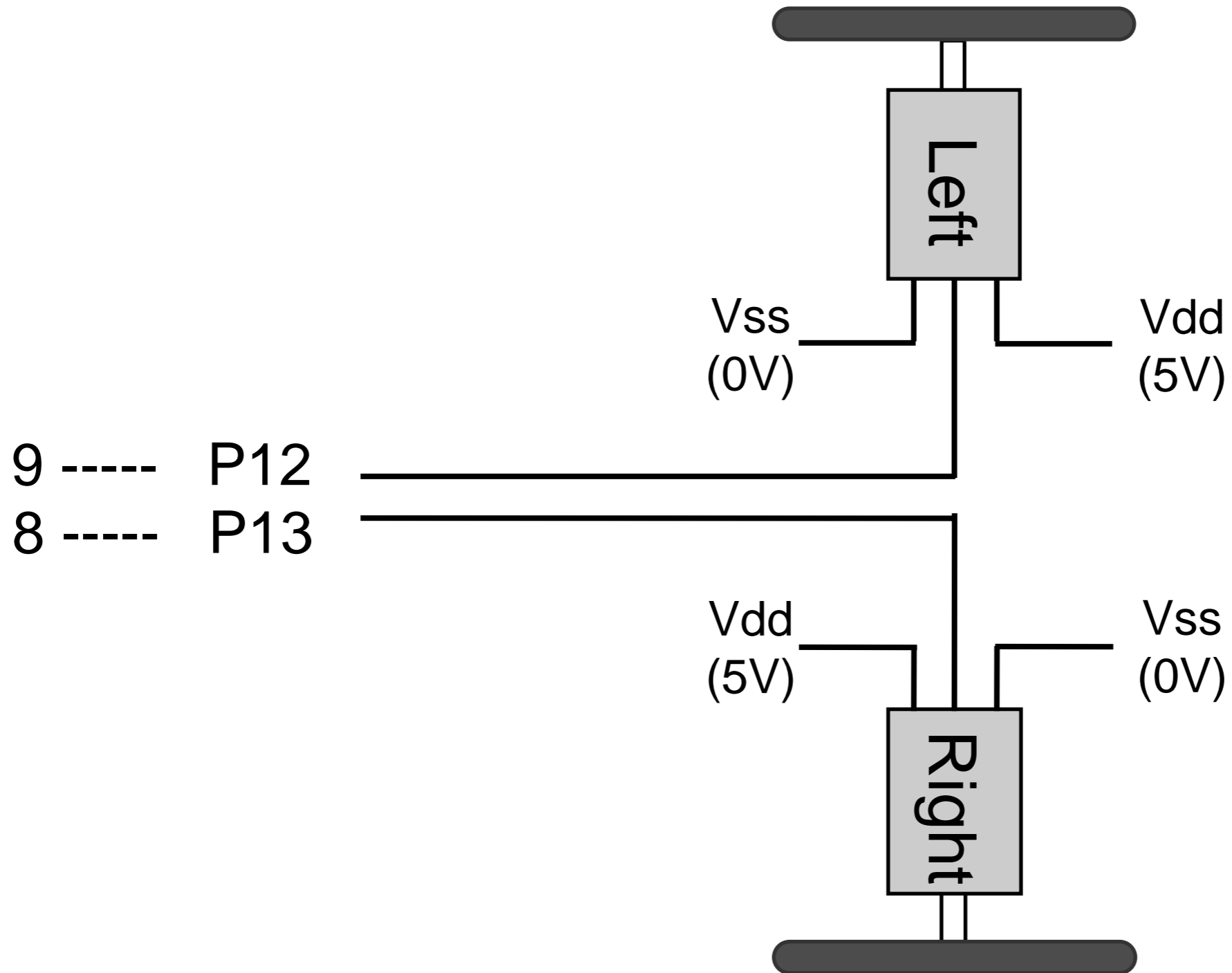
```
long microsecondsToCentimeters(long microseconds)
```

```
{
```

```
    return microseconds / 29 / 2;
```

```
}
```

# Servo Motors





# Motor Command

```
include <Servo.h>
Servo myservo_left;           // create servo object to control left servo
Servo myservo_right;         // create servo object to control right servo
void setup() {
  myservo_left.attach(9);     // attach the servo on pin 9 to the left servo
  myservo_right.attach(8);    // attach the servo on pin 9 to the right servo
}
void loop() {
  myservo_left.write(1000);   // move left motor clockwise
  delay(1000);
  myservo_left.write(1500);   // stop left motor
  delay(1000);
  myservo_left.write(2000);   // move right motor anticlockwise
  delay(1000);
}
```

# Moving Around

	Left	Right
Forward	CCW	CW
Backward	CW	CCW
Left	CW	CW
Right	CCW	CCW