ROCO222: Intro to sensors and actuators

Lecture 4

Arduino RC servo control

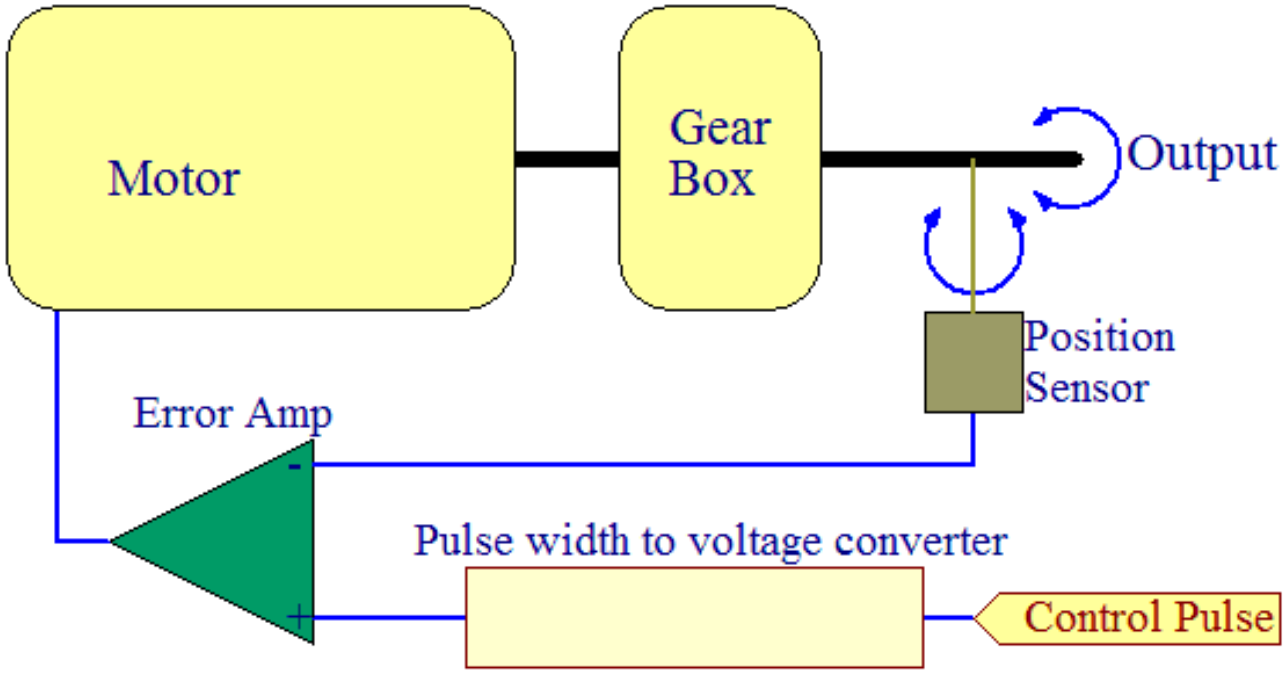
Arduino RC servos

- Small electric motor driving a gear train
- Potentiometer position measurement
- Small
- Cheap

- Servo Motors are electronic devices that convert digital signal to rotational movement.
- Standard servos that their rotation is limited to maximum of 180 degrees in each direction
- Continuous Rotation Servos that can provide rotation unlimitedly in both directions

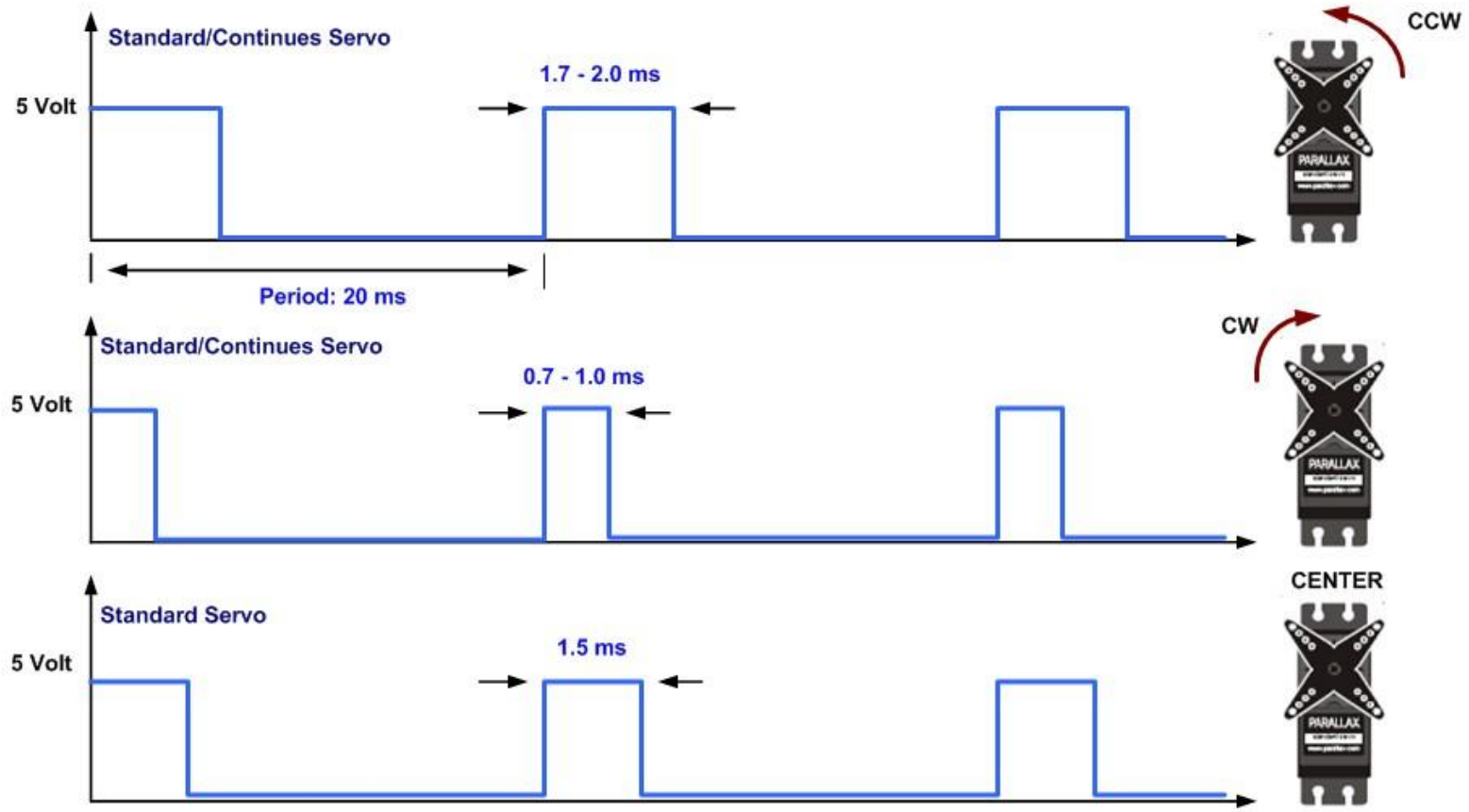
RC servo operation

- Output position compared to the commanded position
- Gives rise to error signal in appropriate direction
- Error drives the electric motor
- When becomes zero servo stops moving



Pulse width control of RC servo

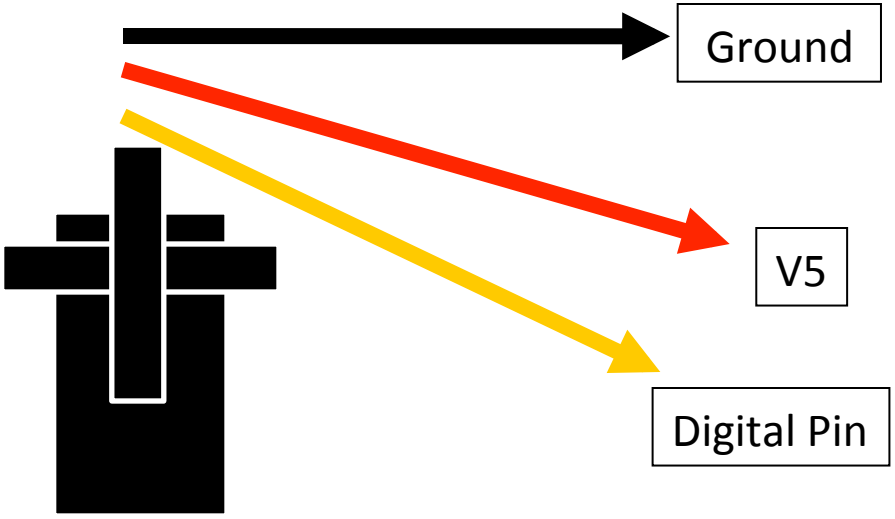
- Commanded signal encoded using PWM
- The pulse duration determines the position of the shaft

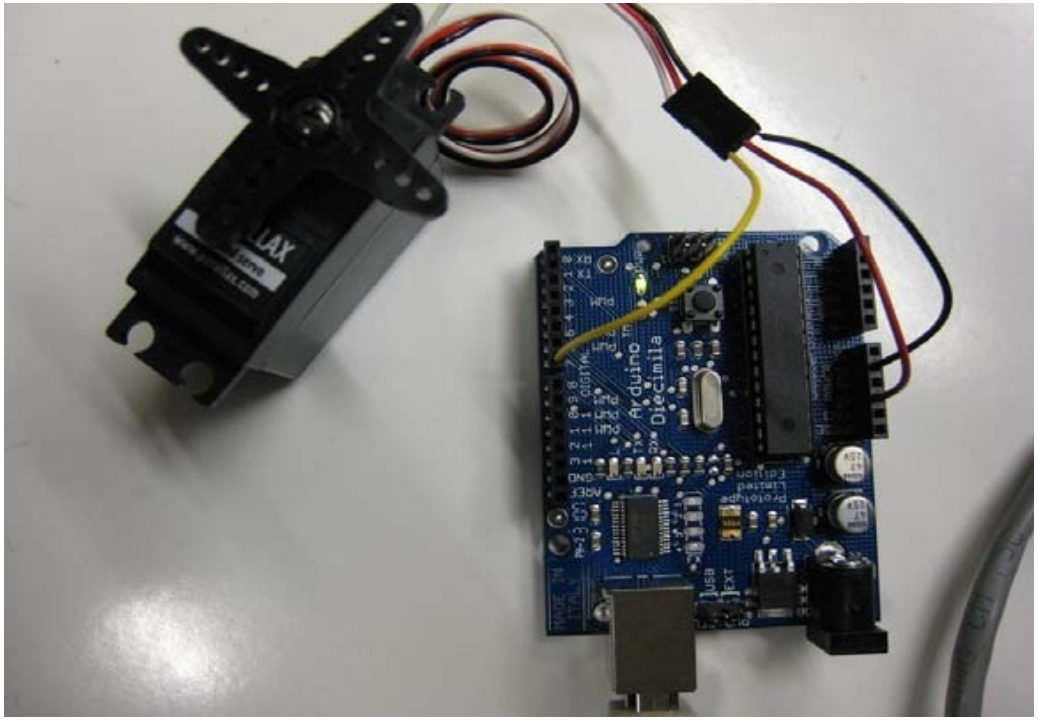


Connecting Arduino to RC servo

Here:

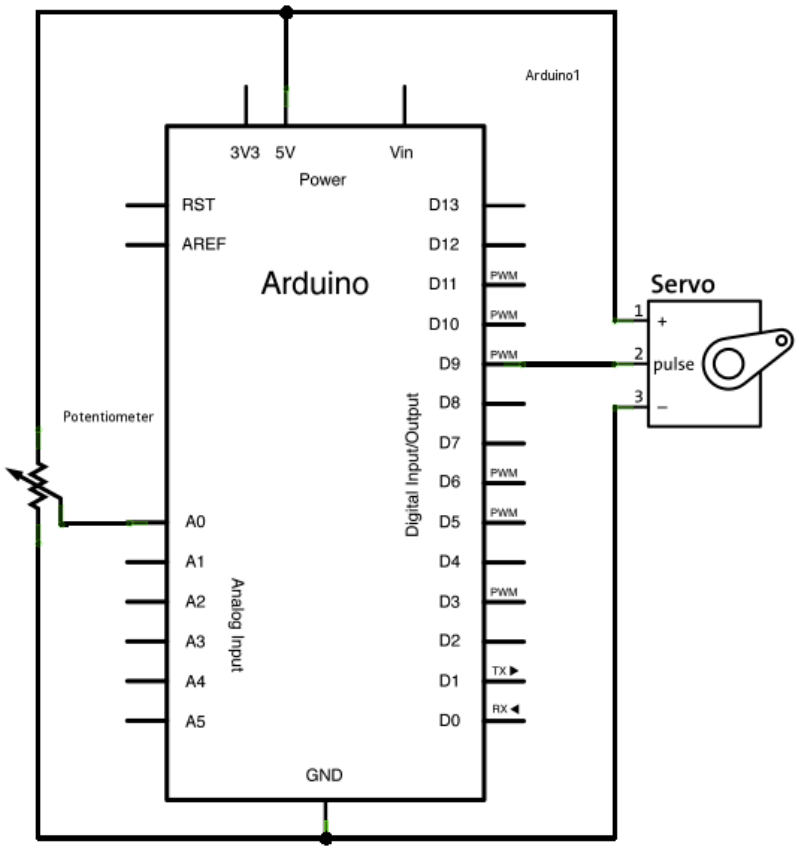
- Black is ground
- Red is connected to 5V
- Yellow wire (sometimes white) is set to the digital pin





Arduino RC servo circuit

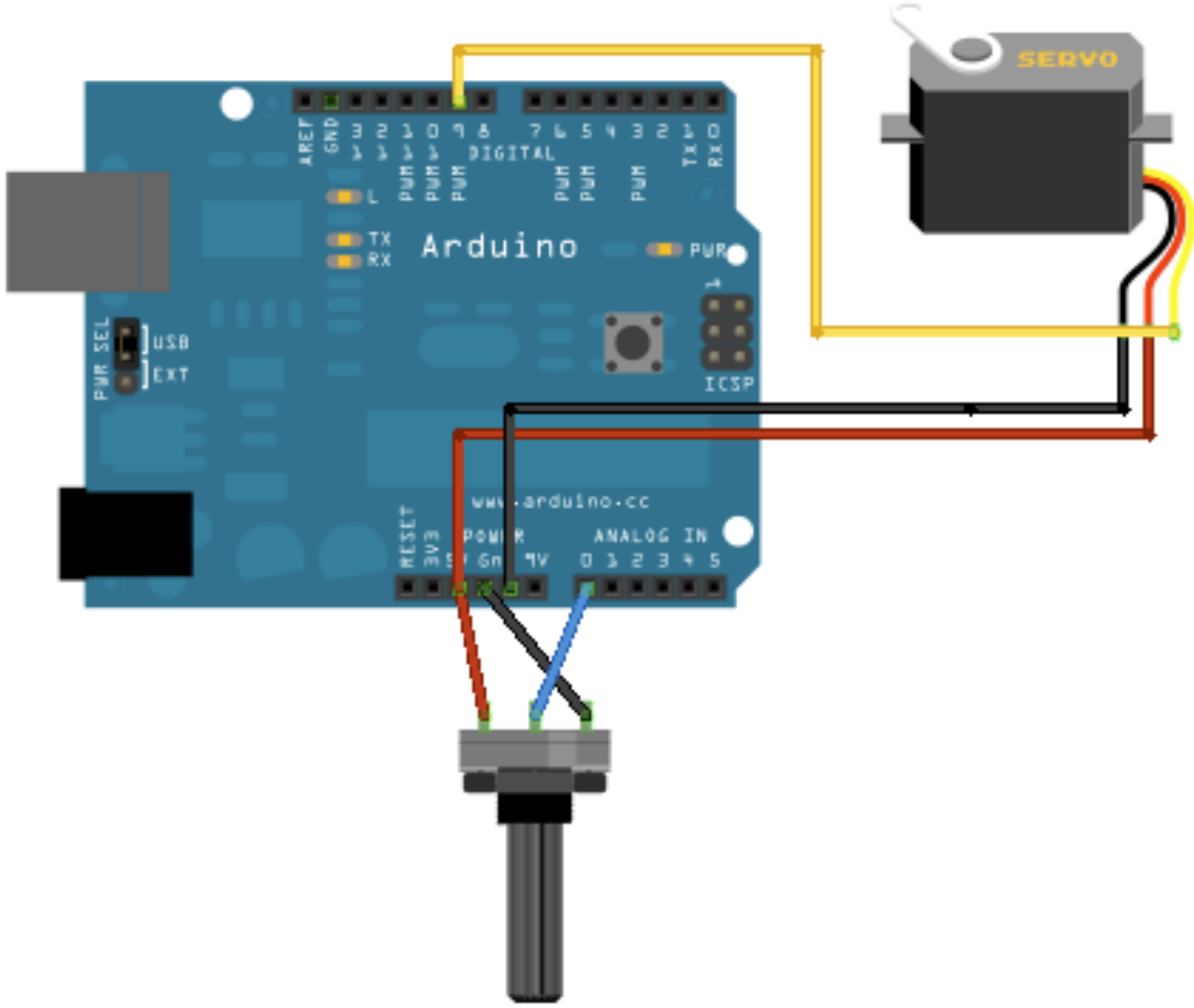
- Servo motors have three wires: power, ground, and signal.
- The power wire is typically red, and should be connected to the 5V pin on the Arduino board.
- The ground wire is typically black or brown and should be connected to a ground pin on the Arduino board.
- The signal pin is typically yellow or orange and should be connected to pin 9 on the Arduino board.
- The potentiometer should be wired so that its two outer pins are connected to power (+5V) and ground, and its middle pin is connected to analog input 0 on the Arduino.



http://arduino.cc/en/Tutorial/Knob

Servo follow-potentiometer example

- Arduino Board
- (1) Servo Motor
- (1) Potentiometer
- hook-up wire



http://arduino.cc/en/Tutorial/Knob

Servo follow-potentiometer example

```
// Controlling a servo position using a potentiometer (variable resistor)
// by Michal Rinott http://people.interaction-ivrea.it/m.rinott
#include <Servo.h>

Servo myservo;  // create servo object to control a servo
int potpin = 0;  // analog pin used to connect the potentiometer
int val;    // variable to read the value from the analog pin

void setup()
{
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}

void loop()
{
  val = analogRead(potpin);            // reads the value of the potentiometer (value between 0 and
1023)
  val = map(val, 0, 1023, 0, 179);     // scale it to use it with the servo (value between 0 and 180)
  myservo.write(val);                  // sets the servo position according to the scaled value
  delay(15);                           // waits for the servo to get there
}
```

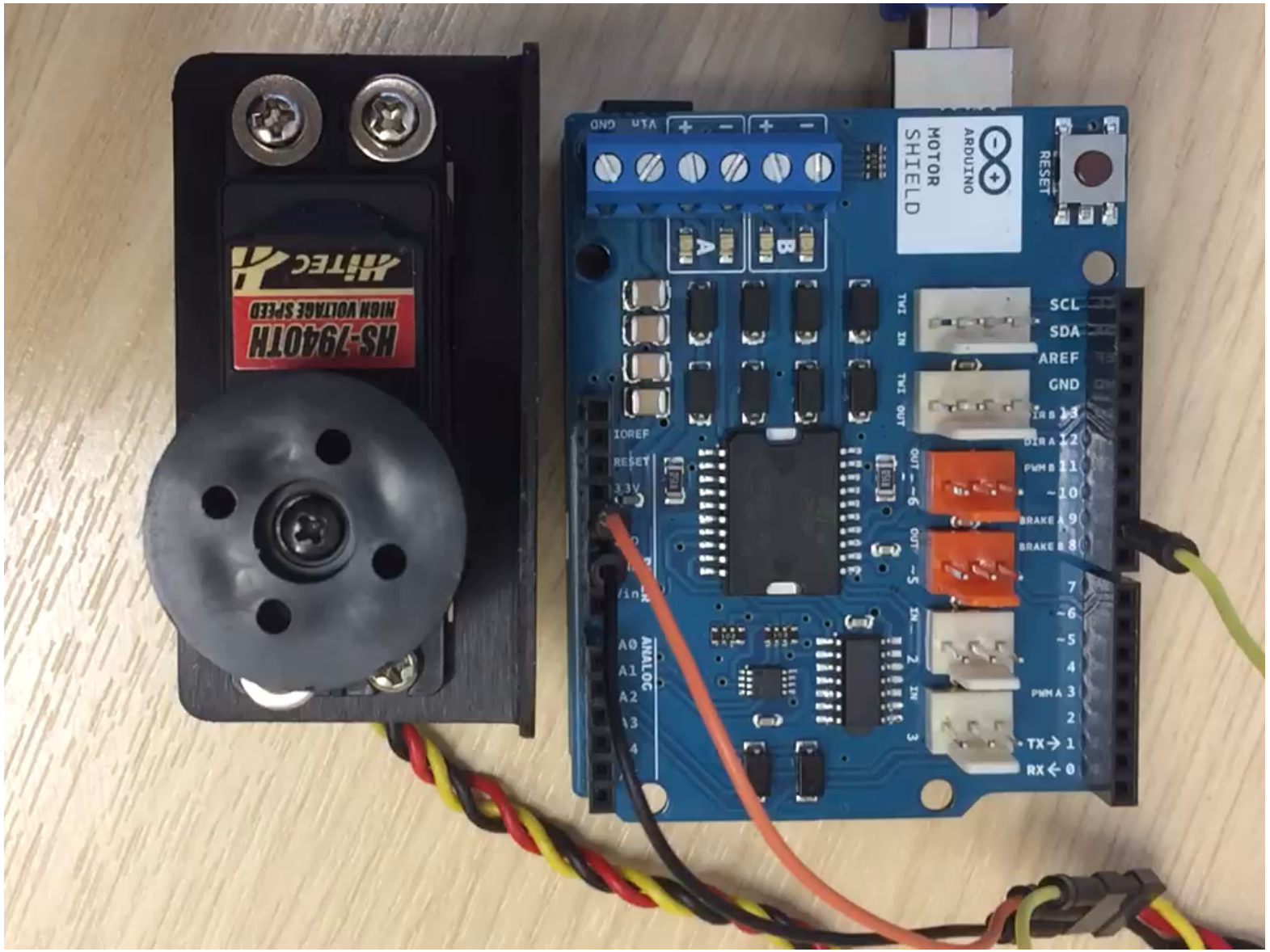
http://arduino.cc/en/Tutorial/Knob

Servo ramp example

```
// Sweep by BARRAGAN http://barraganstudio.com  This example code is in the public domain.
#include <Servo.h>
Servo myservo;                          // create servo object to control a
int pos = 0;                            // variable to store the servo position
void setup()
{
  myservo.attach(9);                    // attaches the servo on pin 9 to the servo object
}

void loop()
{
  for(pos = 0; pos < 180; pos += 1)     // goes from 0 degrees to 180 degrees
  {                                     // in steps of 1 degree
    myservo.write(pos);                             // tell servo to go to position in variable 'pos'
    delay(15);                          // waits 15ms for the servo to reach the position
  }
  for(pos = 180; pos>=1; pos-=1)        // goes from 180 degrees to 0 degrees
  {
    myservo.write(pos);                  // tell servo to go to position in variable 'pos'
    delay(15);                          // waits 15ms for the servo to reach the position
  }
}
```

http://arduino.cc/en/Tutorial/Knob

Sweep code running



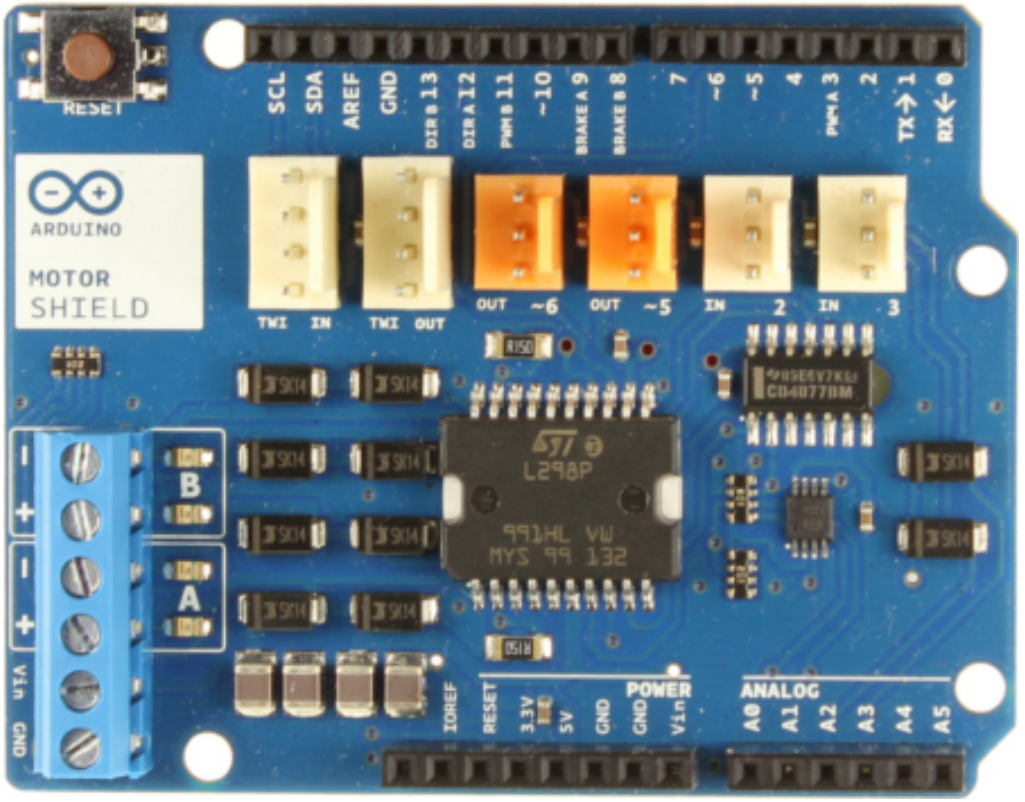
ROCO222: Intro to sensors and actuators

Lecture 4

Arduino motor shield

Arduino motor shield

- Based on the L298 dual full-bridge driver
- designed to drive inductive loads
- Relays
- Solenoids
- DC and stepping motors.
- Drives two DC motors
- Controlling speed
- Direction of each
- Can measure motor current



http://arduino.cc/en/Main/ArduinoMotorShieldR3

Arduino motor shield spec

- Motor controller L298P
- Operating Voltage 5V to 12V
- Max current 2A per channel
  4A max (with external power supply)

- Current sensing 1.65V/A

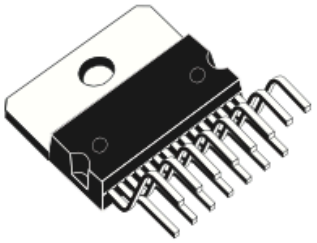
- Drives 2 DC motors or 1 stepper motor
- Free running stop and brake function

L298 dual full-bridge driver

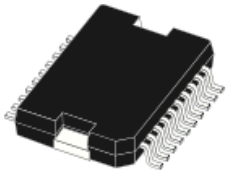
L298

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)



Multiwatt15



PowerSO20

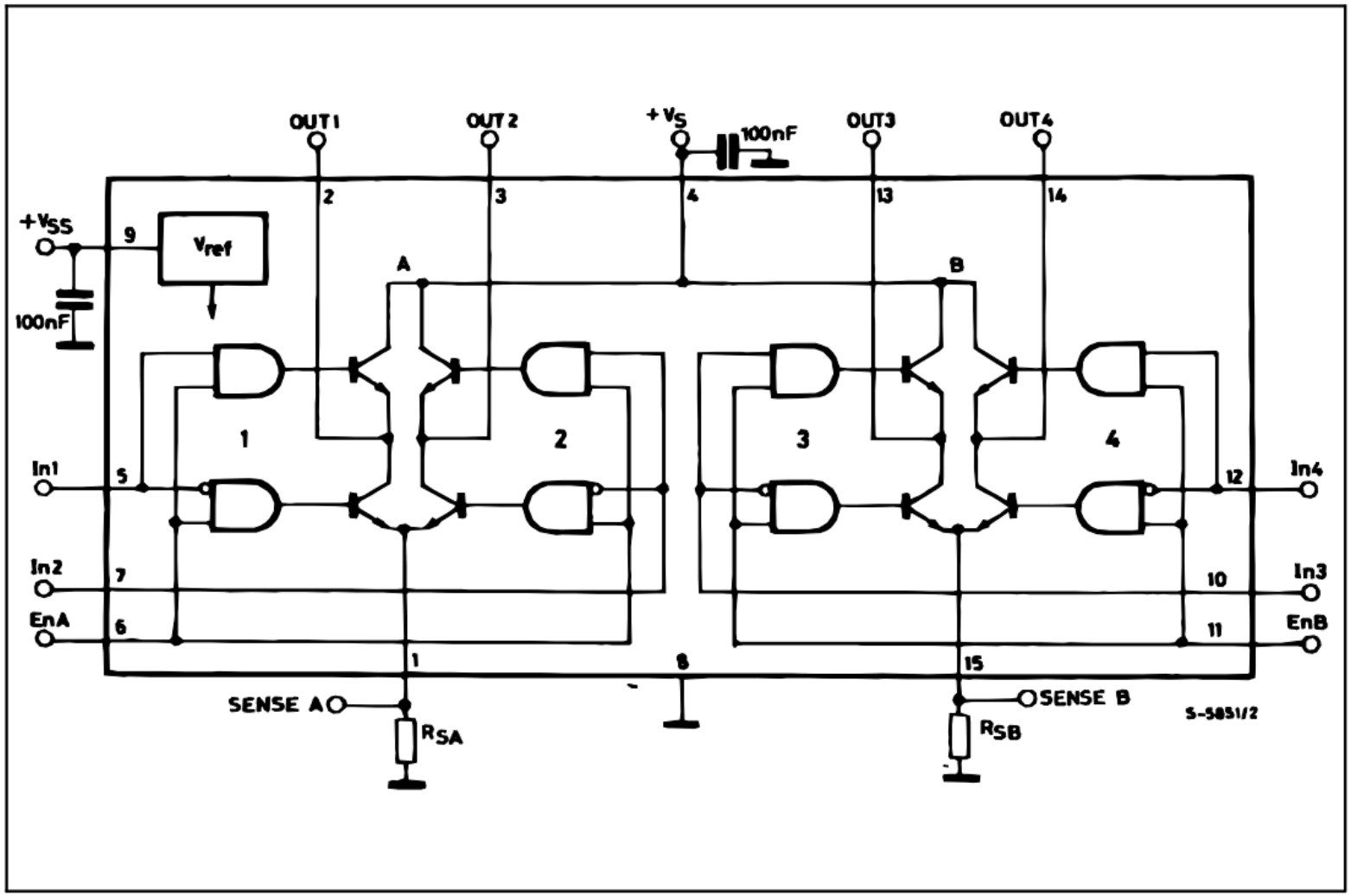
ORDERING NUMBERS : L298N (Multiwatt Vert.)
L298HN (Multiwatt Horiz.)
L298P (PowerSO20)

DESCRIPTION

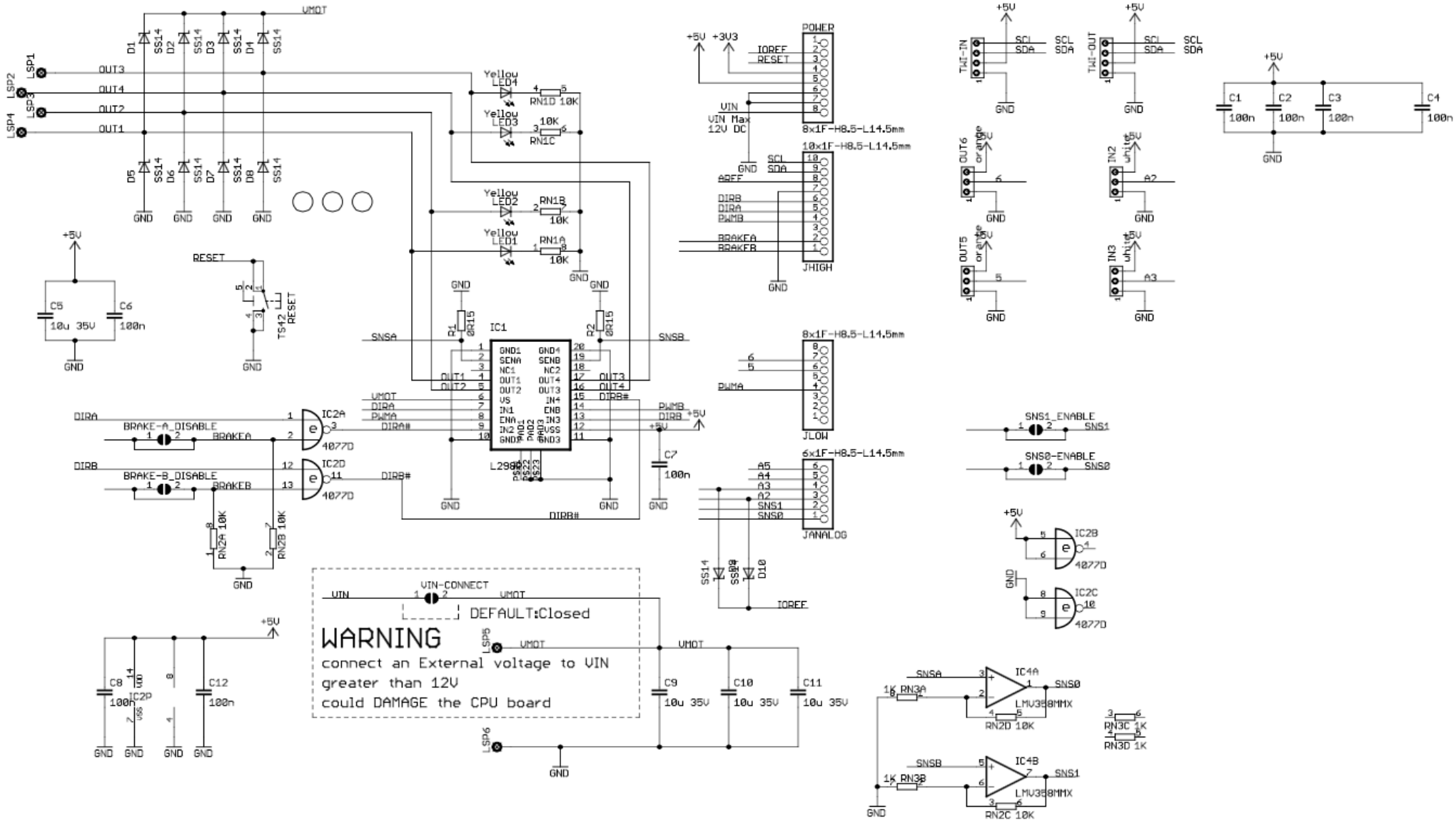
The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

L298 dual full-bridge driver

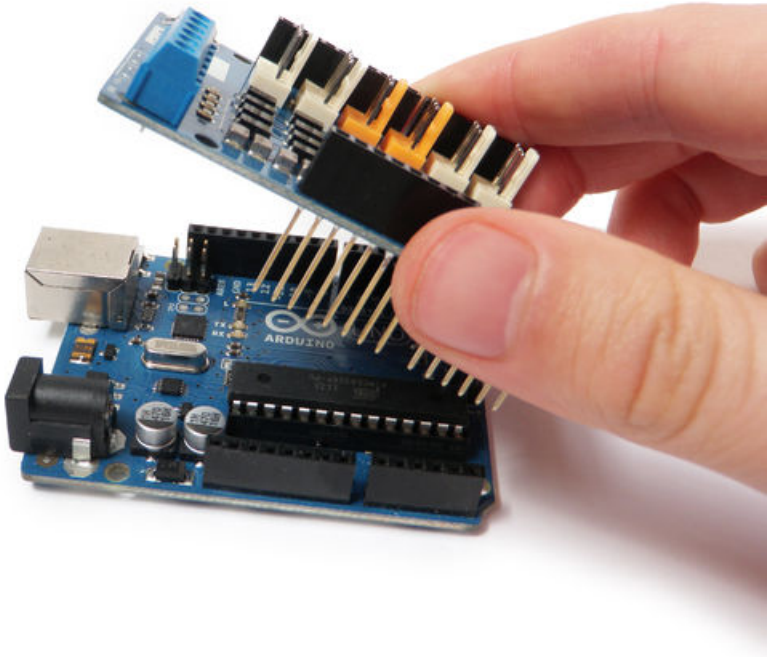
BLOCK DIAGRAM

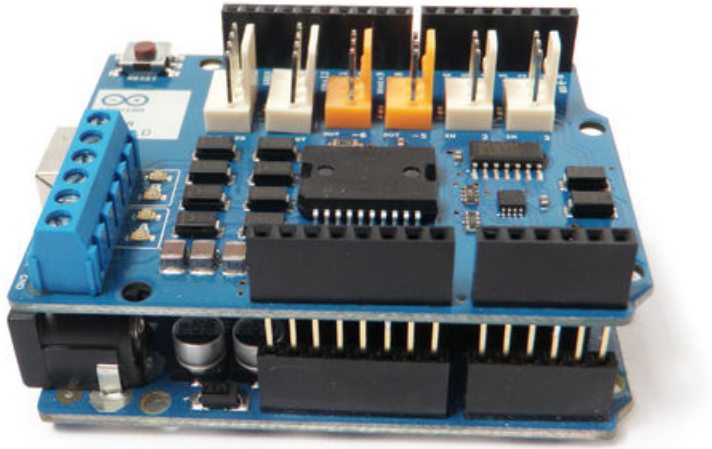


Motor shield schematic



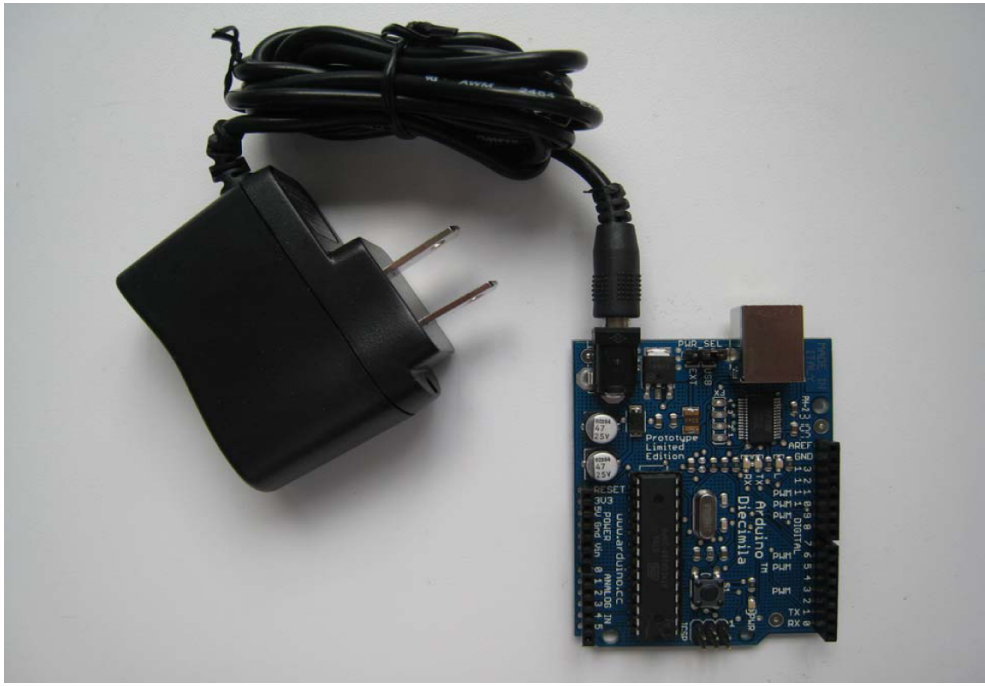
Install the Arduino motor shield





Arduino motor shield power supply

- Required motor current almost always exceeds the maximum USB current rating!
- Must be powered only by an external power supply
- Adapter can be connected by plugging a 2.1mm center-positive plug into the Arduino's board power jack
- Use an external power supply between 7 and 12V



Arduino motor shield output channels

- The motor shield has 2 separate channels: A and B
- Each use 4 of the Arduino pins to drive or sense the motor

- Use each channel separately to drive two DC motors
- Combine them to drive one stepper motor

- Has 6 headers for the attachment of Tinkerkit inputs, outputs, and communication lines.

- With an external power supply, the motor shield can safely supply up to 12V and 2A per motor channel (or 4A to a single channel).

Pins on the Arduino always in use by the motor shield

- By addressing these pins you can
- Select a motor channel to initiate
- Specify the motor direction (polarity)
- Set motor speed (PWM)
- Stop and start the motor
- Monitor the current absorption of each channel

Motor shield pin usage

Function	Channel A	Channel B
Direction	Digital 12	Digital 13
Speed (PWM)	Digital 3	Digital 11
Brake	Digital 9	Digital 8
Current Sensing	Analog 0	Analog 1

ROCO222: Intro to sensors and actuators

Lecture 4

Arduino DC motor control

DC Motors connections

- Drive two Brushed DC motors by connecting the two wires of each one in the (+) and (-) screw terminals for each channel A and B
- Control direction by setting HIGH or LOW the DIR A and DIR B pins
- Control the speed by varying the PWM A and PWM B duty cycle values
- Brake A and Brake B pins set HIGH will brake the DC motors
- Measure DC motor current reading the SNS0 and SNS1 pins
- voltage proportional to the measured current, which can be read as a normal analog input, through the function analogRead() on the analog input A0 and A1.

Motor shield 1-channel DC motor demo

Plug the motor's positive (red) wire into Channel A's + terminal on the motor shield

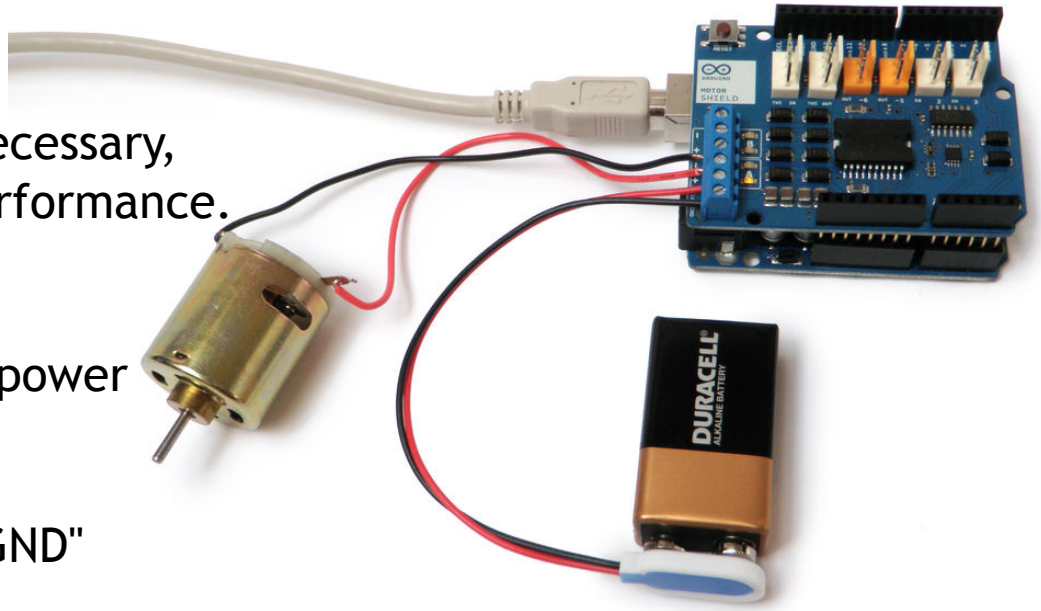
Plug the motor's ground (black) wire into Channel A's - terminal on the shield

An external power supply is not always necessary, but it drastically improves the motor's performance.

To connect your external power supply:
Connect the positive (red) wire from the power supply to the "Vin" terminal

Connect the ground (black) wire to the "GND" terminal.

Upload the code to control the Motor Shield from Arduino.



Motor shield 1-channel DC motor demo

Function	Channel A	Channel B
Direction	Digital 12	Digital 13
Speed (PWM)	Digital 3	Digital 11
Brake	Digital 9	Digital 8
Current Sensing	Analog 0	Analog 1

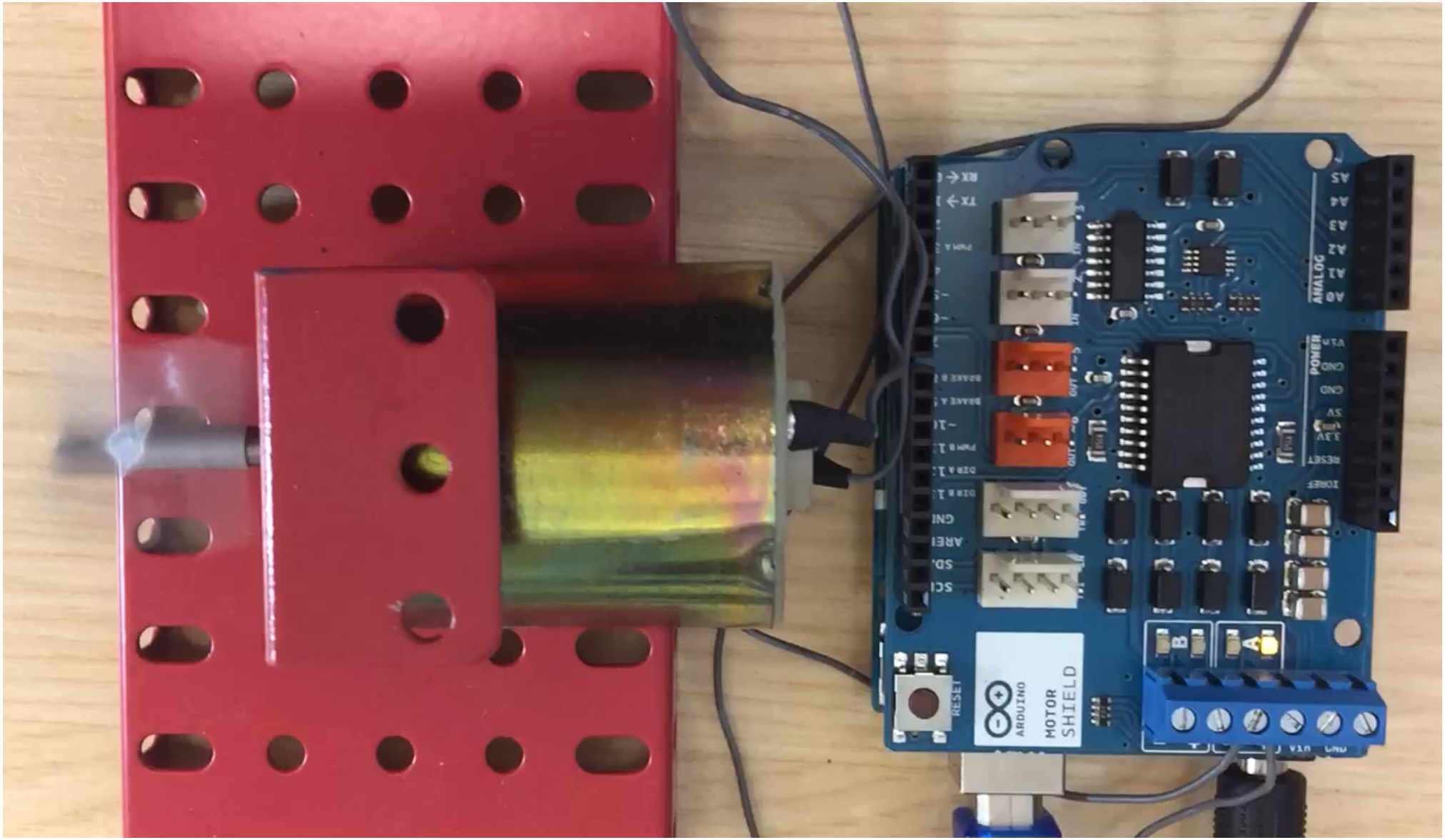
```
/*************************************************************
Motor Shield 1-Channel DC Motor Demo
by Randy Sarafan
For more information see:
http://www.instructables.com/id/Arduino-Motor-Shield-Tutorial
*************************************************************/

void setup() {

  //Setup Channel A
  pinMode(12, OUTPUT);  //Initiates Motor Channel A pin
  pinMode(9, OUTPUT);   //Initiates Brake Channel A pin

}
```

Motor shield 1-channel DC motor demo

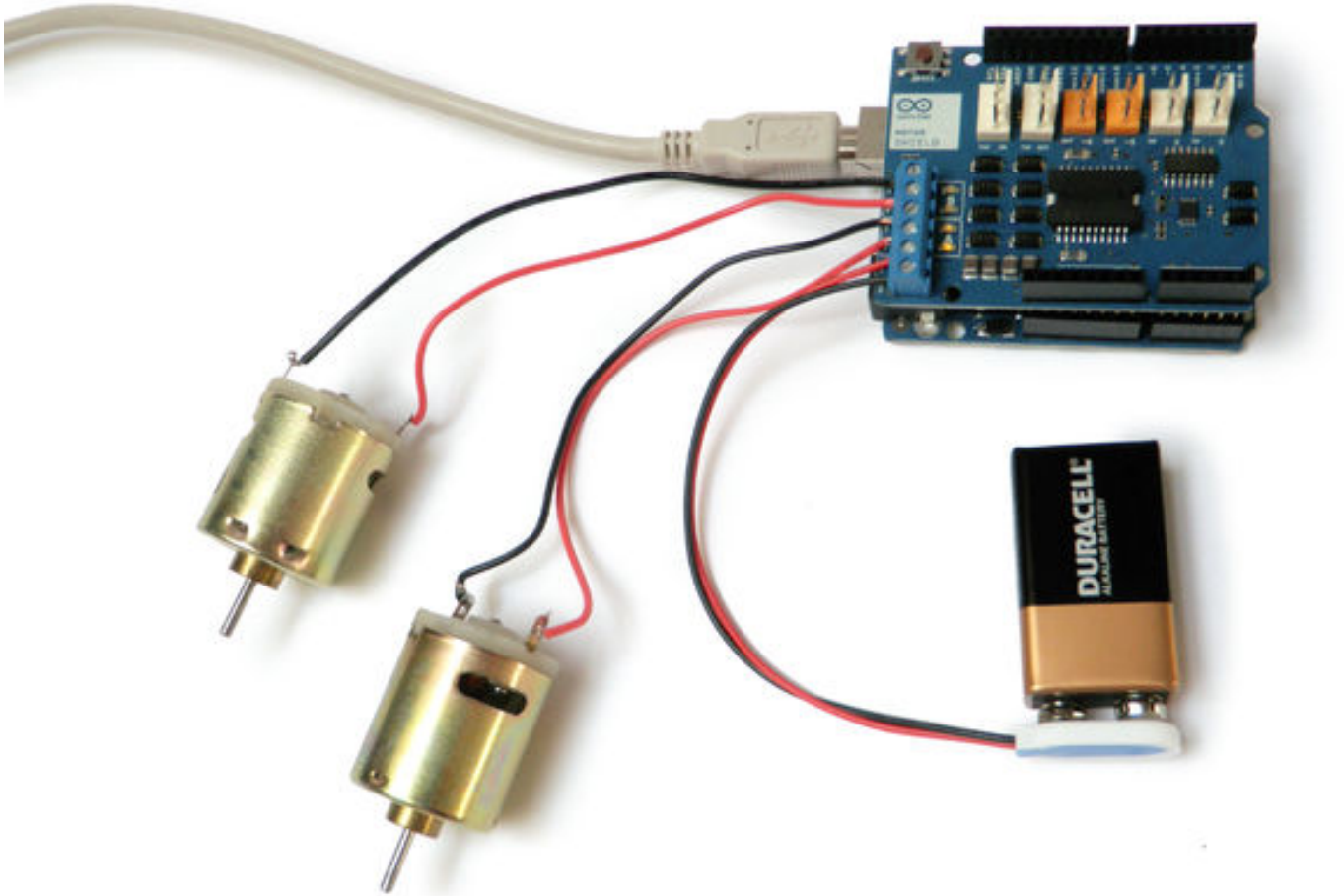
Function	Channel A	Channel B
Direction	Digital 12	Digital 13
Speed (PWM)	Digital 3	Digital 11
Brake	Digital 9	Digital 8
Current Sensing	Analog 0	Analog 1

```
void loop(){

  //forward @ full speed
  digitalWrite(12, HIGH); //Establishes forward direction of Channel A
  digitalWrite(9, LOW);   //Disengage the Brake for Channel A
  analogWrite(3, 255);    //Spins the motor on Channel A at full speed
  delay(3000);

  digitalWrite(9, HIGH); //Engage the Brake for Channel A
  delay(1000);

  //backward @ half speed
  digitalWrite(12, LOW); //Establishes backward direction of Channel A
  digitalWrite(9, LOW);   //Disengage the Brake for Channel A
  analogWrite(3, 123);    //Spins the motor on Channel A at half speed
  delay(3000);

  digitalWrite(9, HIGH); //Engage the Brake for Channel A
  delay(1000);
}
```

DC motor running



Motor shield 2-channel DC motor demo



Motor shield 2-channel DC motor demo

```
/*****************************************************************
Motor Shield 2-Channel DC Motor Demo
by Randy Sarafan

For more information see:
http://www.instructables.com/id/Arduino-Motor-Shield-Tutorial/

*****************************************************************/

void setup() {
  //Setup Channel A
  pinMode(12, OUTPUT); //Initiates Motor Channel A pin
  pinMode(9, OUTPUT); //Initiates Brake Channel A pin

  //Setup Channel B
  pinMode(13, OUTPUT); //Initiates Motor Channel A pin
  pinMode(8, OUTPUT);  //Initiates Brake Channel A pin
}
```

Function	Channel A	Channel B
Direction	Digital 12	Digital 13
Speed (PWM)	Digital 3	Digital 11
Brake	Digital 9	Digital 8
Current Sensing	Analog 0	Analog 1

Motor shield 2-channel DC motor demo

```
void loop(){

  //Motor A forward @ full speed
  digitalWrite(12, HIGH); //Establishes forward direction of Channel A
  digitalWrite(9, LOW);   //Disengage the Brake for Channel A
  analogWrite(3, 255);   //Spins the motor on Channel A at full speed

  //Motor B backward @ half speed
  digitalWrite(13, LOW);  //Establishes backward direction of Channel B
  digitalWrite(8, LOW);   //Disengage the Brake for Channel B
  analogWrite(11, 123);    //Spins the motor on Channel B at half speed
  delay(3000);

  digitalWrite(9, HIGH);  //Engage the Brake for Channel A
  digitalWrite(9, HIGH);  //Engage the Brake for Channel B
  delay(1000);

  //Motor A forward @ full speed
  digitalWrite(12, LOW);  //Establishes backward direction of Channel A
  digitalWrite(9, LOW);   //Disengage the Brake for Channel A
  analogWrite(3, 123);    //Spins the motor on Channel A at half speed

  //Motor B forward @ full speed
  digitalWrite(13, HIGH); //Establishes forward direction of Channel B
  digitalWrite(8, LOW);   //Disengage the Brake for Channel B
  analogWrite(11, 255);   //Spins the motor on Channel B at full speed
delay(3000);

  digitalWrite(9, HIGH);  //Engage the Brake for Channel A
  digitalWrite(9, HIGH);  //Engage the Brake for Channel B
  delay(1000);
}
```

Function	Channel A	Channel B
Direction	Digital 12	Digital 13
Speed (PWM)	Digital 3	Digital 11
Brake	Digital 9	Digital 8
Current Sensing	Analog 0	Analog 1