LCD Display:

```
#include<LiquidCrystal.h>
  LiquidCrystal lcd(12,11,5,4,3,2);//12-RS,11-En,5,4,3,2-Display
void setup() {
  lcd.begin(16,2);//16-no. of col. , 2-no. of row
  lcd.print('HELLO');
   }
void loop() {
  lcd.setCursor(0,1);// 0-col, 1-row
}
```

DC Motor:

```
void setup() {
   pinMode(3,OUTPUT);
   pinMode(5,OUTPUT);
   pinMode(6,OUTPUT);
   digitalWrite(3,HIGH);
}

void loop() {

    digitalWrite(5,HIGH);
    digitalWrite(6,LOW);
    delay(5000);
    digitalWrite(6,HIGH);
    digitalWrite(5,LOW);
    delay(5000);
}
```

#Ultrasonic Sensor:

```
#define trigPin 13
#define echoPin 12
#define led 11
#define led2 10

void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(led, OUTPUT);
  pinMode(led2, OUTPUT);
}

void loop() {
  long duration, distance;
  digitalWrite(trigPin, LOW);  // Added this line
  delayMicroseconds(2); // Added this line
  digitalWrite(trigPin, HIGH);
//  delayMicroseconds(1000); - Removed this line
  delayMicroseconds(10); // Added this line
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;
  if (distance < 4) {  // This is where the LED On/Off happens
    digitalWrite(led,HIGH); // When the Red condition is met, the
Green LED should turn off
  digitalWrite(led2,LOW);
}
```

```
  else {
    digitalWrite(led,LOW);
    digitalWrite(led2,HIGH);
  }
  if (distance >= 200 || distance <= 0){
    Serial.println("Out of range");
  }
  else {
    Serial.print(distance);
    Serial.println(" cm");
  }
  delay(500);
}
```

#Servo Motor:

```
 #include <Servo.h>

int servoPin = 3;
// Create a servo object
Servo Servo1;

void setup() {
  Servo1.attach(servoPin);
}

void loop(){
  // Make servo go to 0 degrees
  Servo1.write(0);
  delay(10000);
  // Make servo go to 90 degrees
```

```
  Servo1.write(90);
  delay(10000);
  // Make servo go to 180 degrees
  Servo1.write(180);
  delay(10000);
}
```

#BUZZER:

```
const int buzzer=9;
void setup()
{
  pinMode(buzzer, OUTPUT);
}

void loop()
{
  tone(buzzer,20000); // frequency in KHz
  delay(2000);
  noTone(buzzer);   //to turn off buzzer
  delay(2000);

}
```

#PIR:

```
//the time we give the sensor to calibrate (10-60 secs according to
the datasheet)
int calibrationTime = 30;

//the time when the sensor outputs a low impulse
```

```
long unsigned int lowIn;

//the amount of milliseconds the sensor has to be low
//before we assume all motion has stopped
long unsigned int pause = 5000;

boolean lockLow = true;
boolean takeLowTime;

int pirPin = 3;    //the digital pin connected to the PIR sensor's
output
int ledPin = 13;



/////////////////////////////
//SETUP
void setup(){
  Serial.begin(9600);
  pinMode(pirPin, INPUT);
  pinMode(ledPin, OUTPUT);
  digitalWrite(pirPin, LOW);

  //give the sensor some time to calibrate
  Serial.print("calibrating sensor ");
    for(int i = 0; i < calibrationTime; i++){
      Serial.print(".");
      delay(1000);
      }
    Serial.println(" done");
    Serial.println("SENSOR ACTIVE");
    delay(50);
```

```
  }


////////////////////////////

//LOOP

void loop(){


     if(digitalRead(pirPin) == HIGH){

       digitalWrite(ledPin, HIGH);   //the led visualizes the sensors
output pin state

       if(lockLow){

         //makes sure we wait for a transition to LOW before any
further output is made:

         lockLow = false;

         Serial.println("---");

         Serial.print("motion detected at ");

         Serial.print(millis()/1000);

         Serial.println(" sec");

         delay(50);

         }

         takeLowTime = true;

       }


     if(digitalRead(pirPin) == LOW){

       digitalWrite(ledPin, LOW);  //the led visualizes the sensors
output pin state


       if(takeLowTime){

        lowIn = millis();          //save the time of the transition
from high to LOW

        takeLowTime = false;       //make sure this is only done at
the start of a LOW phase

        }
```

```
       //if the sensor is low for more than the given pause,
       //we assume that no more motion is going to happen
       if(!lockLow && millis() - lowIn > pause){
           //makes sure this block of code is only executed again
after
           //a new motion sequence has been detected
           lockLow = true;
           Serial.print("motion ended at ");      //output
           Serial.print((millis() - pause)/1000);
           Serial.println(" sec");
           delay(50);
           }
       }
  }
```

LDR:

```
void setup() {
 pinMode(A0,INPUT);
 Serial.begin(9600);
}
void loop() {
 int value=analogRead(A0);
 Serial.println(value);
 delay(1000);
}
```

#LM 35

```
void setup() {
  pinMode(A0,INPUT);
  Serial.begin(9600);
}

void loop() {
  float value=analogRead(A0);
  Serial.println(value*0.48);
  delay(1000);
}
```

#Push Button: