Bluetooth:

```
#define ledPin 13
int state = 0;
void setup() {
  pinMode(ledPin, OUTPUT);
  digitalWrite(ledPin, LOW);
  Serial.begin(9600); // Default communication rate of the Bluetooth module
}
void loop() {
  if(Serial.available() > 0){ // Checks whether data is comming from the serial port
    state = Serial.read(); // Reads the data from the serial port
 }
 if (state == '0') {
  digitalWrite(ledPin, LOW); // Turn LED OFF
  Serial.println("LED: OFF"); // Send back, to the phone, the String "LED: ON"
  state = 0;
 }
 else if (state == '1') {
  digitalWrite(ledPin, HIGH);
  Serial.println("LED: ON");
  state = 0;
 }
}
```

DC Motor:

```
int LeftMotorForward = 7; // Pin 7 has Left Motor connected on Arduino boards.
int LeftMotorReverse = 6; // Pin 6 has Left Motor connected on Arduino boards.
void setup()
{
 Serial.begin(9600);
 pinMode(LeftMotorForward, OUTPUT);
 pinMode(LeftMotorReverse, OUTPUT);
}

void loop() {
  // Forward
  digitalWrite(LeftMotorForward, HIGH);
  digitalWrite(LeftMotorReverse, LOW);
  delay(2000);

  // Reverse
  digitalWrite(LeftMotorReverse, HIGH);
  digitalWrite(LeftMotorForward, LOW);
  delay(2000);

  // Stop
  digitalWrite(LeftMotorReverse, LOW);
  digitalWrite(LeftMotorForward, LOW);
  delay(2000);
}
```

KeyPadAddition:

```
#include <Keypad.h>
```

```
const byte ROWS = 4; // Four rows

const byte COLS = 4; // Three columns

// Define the Keymap

char keys[ROWS][COLS] = {

  {'1', '2', '3', '/'},

  {'4', '5', '6', '*'},

  {'7', '8', '9', '-'},

  {'C', '0', '=', '+'}

};

// Connect keypad ROW0, ROW1, ROW2 and ROW3 to these Arduino pins.

byte rowPins[ROWS] = { 4, 5, 6, 7 };

// Connect keypad COL0, COL1 and COL2 to these Arduino pins.

byte colPins[COLS] = { 8, 9, 10, 11};


// Create the Keypad

Keypad kpd = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );


boolean presentValue = false;

boolean final = false;

String num1;

String num2;

int answer;

char op;


void setup()

{
  Serial.begin(9600);
}

void loop() {
  char key = kpd.getKey();

  if (key == '1' || key == '2' || key == '3' || key == '4' || key == '5' || key == '6' || key == '7' || key == '8' || key == '9' || key == '0')
  {
    if (presentValue == false)
    {
      num1 = num1 + key;
      Serial.print(key);
    }
    else
    {
      num2 = num2 + key;
      Serial.print(key);
      final = true;
    }
  }

  else if (presentValue == false  && (key =='+'))
  {

      presentValue = true;
      op = key;
      Serial.print(op);

  }

  else if (final == true && key == '=') {

    answer = num1.toInt() + num2.toInt();

    Serial.println();
    Serial.println(answer);
    presentValue = false;
    final = false;
    num1 = "";
    num2 = "";
    answer = 0;
    op = ' ';
  }
  else if (key == 'C') {
    //Serial.clear();
    presentValue = false;
    final = false;
    num1 = "";
    num2 = "";
    answer = 0;
    op = ' ';
  }
}
```

KeyPad Print:

```
#include <Keypad.h>


const byte ROWS = 4; // Four rows
const byte COLS = 4; // Three columns
// Define the Keymap
char keys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
// Connect keypad ROW0, ROW1, ROW2 and ROW3 to these Arduino pins.
byte rowPins[ROWS] = { 2,3,4,5 };
// Connect keypad COL0, COL1 and COL2 to these Arduino pins.
byte colPins[COLS] = { 6,7,8,9};


// Create the Keypad
Keypad kpd = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );


void setup()
{
  Serial.begin(9600);
}


void loop()
{
  char key = kpd.getKey();
  if(key)  // Check for a valid key.
  {
       Serial.println(key);
   }
}
```

KeyPadTest:

```
#include <Keypad.h>

const byte ROWS = 4; // Four rows
const byte COLS = 4; // Three columns
// Define the Keymap
char keys[ROWS][COLS] = {
 {'1','2','3','A'},
 {'4','5','6','B'},
 {'7','8','9','C'},
 {'*','0','#','D'}
};
// Connect keypad ROW0, ROW1, ROW2 and ROW3 to these Arduino pins.
byte rowPins[ROWS] = { 4,5,6,7 };
// Connect keypad COL0, COL1 and COL2 to these Arduino pins.
byte colPins[COLS] = { 8,9,10,11};

// Create the Keypad
Keypad kpd = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
```

```
}

#define ledpin 13

void setup()
{
  pinMode(ledpin,OUTPUT);
  digitalWrite(ledpin, HIGH);
  Serial.begin(9600);
}

void loop()
{
  char key = kpd.getKey();
  if(key)  // Check for a valid key.
  {
    switch (key)
    {
      case '*':
        digitalWrite(ledpin, LOW);
        break;
      case '#':
        digitalWrite(ledpin, HIGH);
        break;
      default:
        Serial.println(key);
    }
  }
}
```

LCD Display:

```
#include <LiquidCrystal.h>
LiquidCrystal LCD(13,12,11,10,9, 8);

void setup() {
  // set up the LCD's number of columns and rows:
  LCD.begin(16, 2);
  // Print a message to the LCD.
  LCD.print("hello, world!");
}

void loop() {
  // Turn off the display:
  LCD.noDisplay();
  delay(500);
  // Turn on the display:
  LCD.display();
  delay(500);
}
```

LCD Function:

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(13,12,11,10,9, 8);
void setup() {
 lcd.begin(16,2); // set up the LCD's number of columns and rows: }
}
void loop() {
 lcd.print("AUST"); // Prints "Arduino" on the LCD
 delay(1000); // 3 seconds delay

 lcd.setCursor(2,1); // Sets the location at which subsequent text written to the LCD will be displayed

 lcd.print("CSE");

 delay(1000);

 lcd.clear(); // Clears the display

 lcd.blink(); //Displays the blinking LCD cursor

 delay(1000);

 lcd.setCursor(7,1);

 delay(1000);

 lcd.noBlink(); // Turns off the blinking LCD cursor

 lcd.cursor(); // Displays an underscore (line) at the position to which the next character will be written

 delay(1000);

 lcd.noCursor(); // Hides the LCD cursor

 lcd.clear(); // Clears the LCD screen

}
```

LDR:

```
int ldr=A4;

int value = 0;

void setup() {

Serial.begin(9600);

}

void loop() {

  value= analogRead(ldr);

  Serial.println("Intensity of the LDR is =");

  Serial.println(value);
  delay(1000);
}
```

One LED :

```
int ledpin = 13;

void setup() {
        pinMode(ledpin,OUTPUT);
}

void loop() {
        digitalWrite(ledpin,HIGH);
        delay(500);
        digitalWrite(ledpin,LOW);
        delay(500);
}
```

Servo:

```
#include <Servo.h>

Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards

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 <= 360; pos=pos+1) { // goes from 0 degrees to 360 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(20);                       // waits 20ms for the servo to reach the position
  }
  for (pos = 360; pos >= 0; pos=pos-1) { // goes from 360 degrees to 0 degrees
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(20);                       // waits 20ms for the servo to reach the position
  }
}
```