Sten-SLATE ESP Kit
Conditional Programming

Stensat Group LLC, Copyright 2020
Conditional Programming
Conditional Programming

Conditional programming is how programs make decisions. The simplest form is the `if()` statement. The `if()` statement requires an argument. The argument is a comparison that results as true or false. False has a value of zero.

The following are valid comparisons:

- `a == b` true if `a` equals `b`
- `a > b` true if `a` greater than `b`
- `a >= b` true if `a` greater or equal to `b`
- `a < b` true if `a` less than `b`
- `a <= b` true if `a` less than or equal to `b`
In this example, the program will monitor the light level and turn on the LED when it gets dark enough.

Reuse the wiring of the light sensor as before. Add a jumper to connect D14 to LED1. Enter the code below and run it. The voltage threshold may need to be adjusted depending on the light level in the room. Adjust it so that the LED is off with the ambient light. Then cover the light sensor. The LED should turn on.

At the top of the code is a `#define` statement. This assigns a constant to a name. This is useful if the constant is used in multiple places. It allows easier changes by only changing the constant in one location. It also helps to use a name that has meaning for its use.

Adjust the `THRESHOLD` constant until the program operates properly.

```c
#define THRESHOLD 500

void setup()
{
  Serial.begin(115200);
  pinMode(14, OUTPUT);
}

void loop()
{
  int a;
  a = analogRead(0);
  Serial.println(a);
  if(a < THRESHOLD) {
    digitalWrite(14, HIGH);
  } else {
    digitalWrite(14, LOW);
  }
  delay(200);
}
```
Reconnect the thermistor to the ADC. Connect all three LEDs to digital pins. Create a program to do the following:

Turn on the green LED when at room temperature

Turn on the yellow LED when it is a bit warmer. Select the threshold.

Turn on the red LED when it is hot. Select the threshold.

Turn off the red LED when it is below the hot threshold.

Turn off the yellow LED when it is below the warm threshold.
Multiple Conditions

Multiple conditions can be tested simultaneously with an if() statement. The first example shows two comparisons with an AND comparison which is represented by &&. The if statement reads as follows: if a equals 45 and b equals c execute the following. Both variable comparisons have to be true for the if() statement result to be true.

The second example shows an OR comparison. The if() statement result is true if either comparisons are true or both comparisons are true. The if statement reads as follows: If a equals 45 or b is greater than c then execute the following. At least one of the comparisons have to be try for the if() statement result to be true.

More than two comparisons can be included in the if() statement. A statement can have several comparisons and all have to be true for the if() statement result to be true using the AND comparison. An if() statement with several comparisons can have at least one comparison to be true using the OR comparison. A combination of AND and OR comparisons can be performed. Care has to be taken on how the comparisons are grouped. The last example shows a combination. Notice the OR comparison has an extra pair of parentheses around it. This tells the compiler to first perform if b is greater than c or b equals 32. The result of that comparison is then compared with if a equals 45. If the results of both are true, the if() statement is true.
The code to the right works the same as the third example in the previous page. It shows the `if()` statement broken down into multiple `if()` statements. This code may be easier to understand.

For an exercise, write a program that requires testing two sensors, the light sensor and ultrasonic sensor. Write a program that turns on the red LED when the room is dark and the ultrasonic sensor is within 20 cm of an object otherwise turn off the red LED. Turn the green LED on when the room is not dark and the ultrasonic sensor is within 20 cm of an object otherwise turn the green LED off.

```cpp
void setup() {
  Serial.begin(115200);
  int a = 45;
  int b = 32;
  int c = 32;
}

void loop() {
  if((b > c) || (b == 32)) {
    if(a == 45) {
      Serial.println("mostly true");
    }
  }
}

AND and OR Comparison Example