Microcontrollers & Applications

Lecture 4.2: Conditionals

If Conditionals (C & Python)

- If the given condition is true, then the statement runs.
  - if (condition) statement;

- If the given condition is true, then the statement#1 runs, otherwise, statement#2 runs.
  - if (condition) statement#1;
  - else statement#2;

- If the given condition#1 is true, then the statement#1 runs, otherwise, if the given condition#2 is true, then the statement#2 runs, otherwise, statement#3 runs.
  - if (condition#1) statement#1;
  - elseif (condition#2) statement#2;
  - else statement#3;

```
if condition:
    statement
```

```
if condition:
    statement#1
else:
    statement#2
```

```
if condition#1:
    statement#1
elif condition#2:
    statement#2
else:
    statement#3
```

Case Conditionals (C & Python)

- If variable_name == val#1, then the statement#1 runs,
- Else if variable_name == val#2, then the statement#2 runs,
- Else if variable_name == val#3, then the statement#3 runs,
- Else if variable_name == val#4, then the statement#4 runs.

```
switch (variable_name) {
    case val#1: statement#1; break;
    case val#2: statement#2; break;
    case val#3: statement#3; break;
    default: statement#N;
    }
```

```
match variable_name:
    case val#1: statement#1
    case val#2: statement#2
    case val#3: statement#3
    case _: statement#N
```

? Conditionals (C)

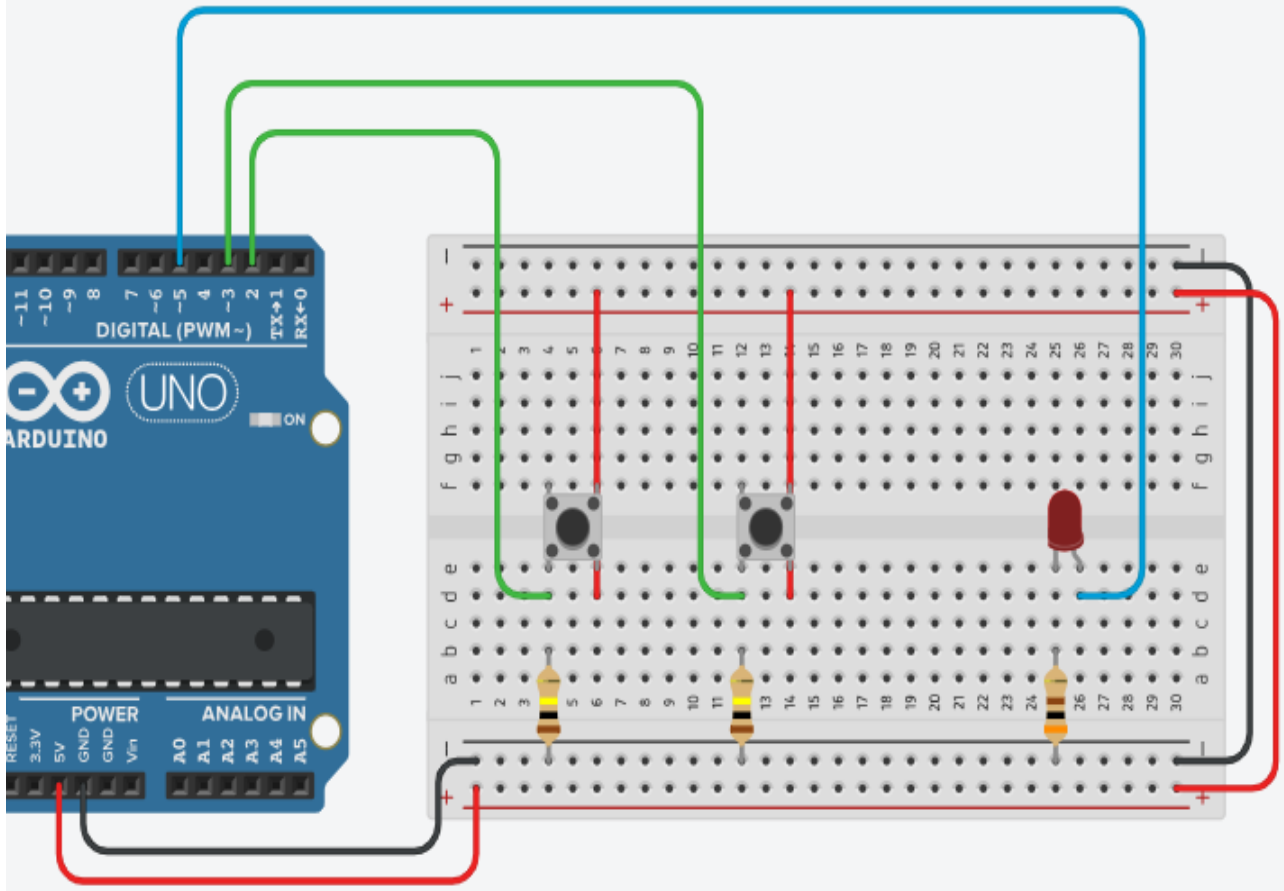
- exp#1 ? exp#2 : exp#3 ;

- If exp#1 is true then the exp#2 runs, else exp#3 runs.

if (exp#1) exp#2;

else exp#3;

Conditional Examples #1

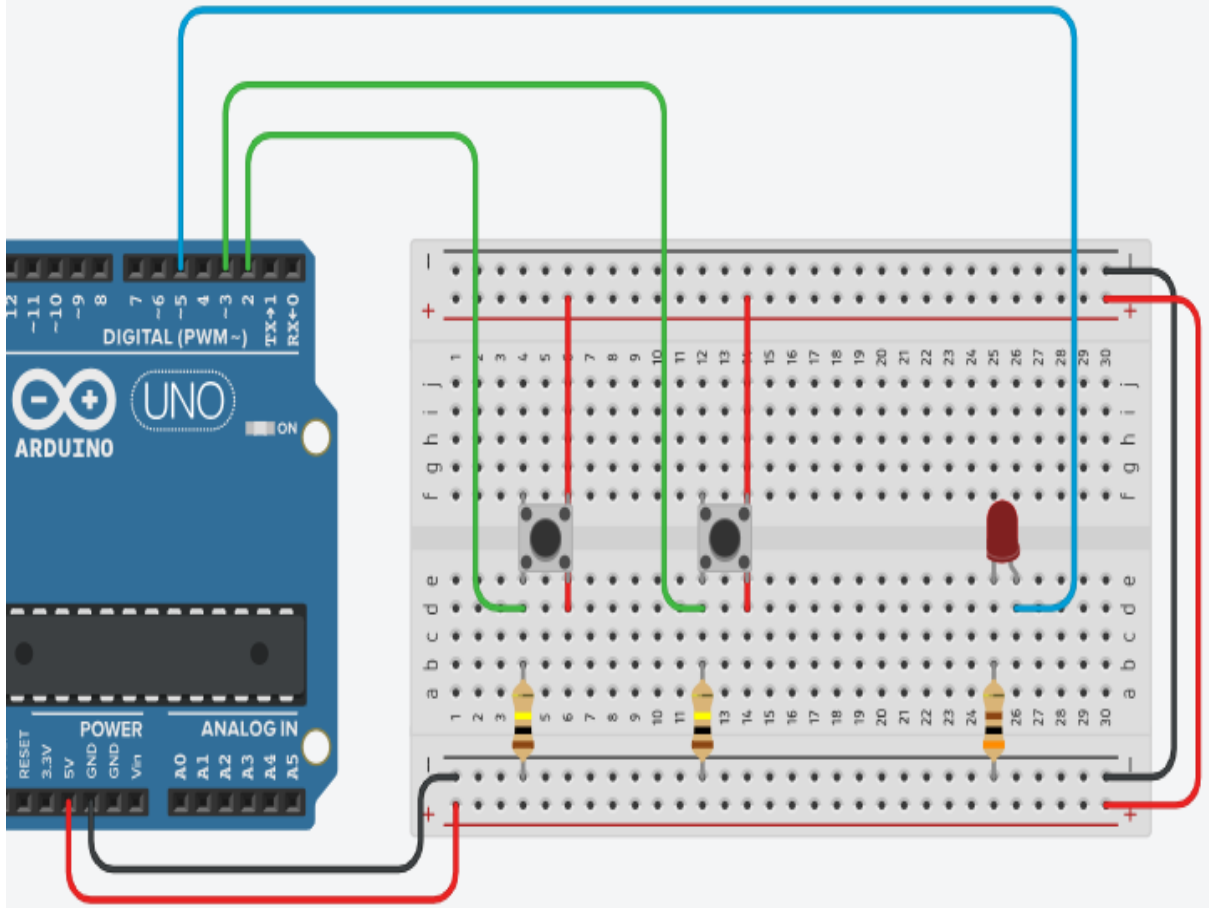



```
1  #define BUTTON_1    2
2  #define BUTTON_2    3
3
4  #define LED  5
5
6  void setup()
7  {
8    pinMode(BUTTON_1, INPUT);
9    pinMode(BUTTON_2, INPUT);
10   pinMode(LED, OUTPUT);
11   digitalWrite(LED, LOW);
12
13   Serial.begin(9600);
14 }
15
16 void loop()
17 {
18   if (digitalRead(BUTTON_1) == HIGH) {
19     digitalWrite(LED, HIGH);
20     Serial.println("LED is ON now");
21     delay(10); // Wait for 10 millisecond(s)
22   }
23   if (digitalRead(BUTTON_2) == HIGH) {
24     digitalWrite(LED, LOW);
25     Serial.println("LED is OFF now");
26     delay(10); // Wait for 10 millisecond(s)
27   }
28 }
```

Serial Monitor

```
LED is ON now
LED is ON now
LED is ON now
LED is ON now
LED is ON now
LED is ON now
LED is ON now
LED is ON now
LED is ON now
LED is ON now
LED is ON now
LED is ON now
LED is ON now
LED is ON now
LED is ON now
LED is OFF now
LED is OFF now
LED is OFF now
LED is OFF now
LED is OFF now
LED is OFF now
LED is OFF now
LED is OFF now
LED is OFF now
LED is OFF now
LED is OFF now
LED is OFF now
LED is OFF now
```

Conditional Examples #2




```
1  #define BUTTON_1    2
2  #define BUTTON_2    3
3
4  #define LED  5
5  int ledStatus = LOW;
6
7  void setup()
8  {
9    pinMode(BUTTON_1, INPUT);
10   pinMode(BUTTON_2, INPUT);
11   pinMode(LED, OUTPUT);
12   digitalWrite(LED, LOW);
13
14   Serial.begin(9600);
15 }
16
17 void loop()
18 {
19   if (digitalRead(BUTTON_1) == HIGH && !ledStatus) {
20     digitalWrite(LED, HIGH);
21     ledStatus = HIGH;
22     Serial.println("LED is ON now");
23   }
24   if (digitalRead(BUTTON_2) == HIGH && ledStatus) {
25     digitalWrite(LED, LOW);
26     ledStatus = LOW;
27     Serial.println("LED is OFF now");
28   }
29 }
```

Serial Monitor

```
LED is ON now
LED is OFF now
```

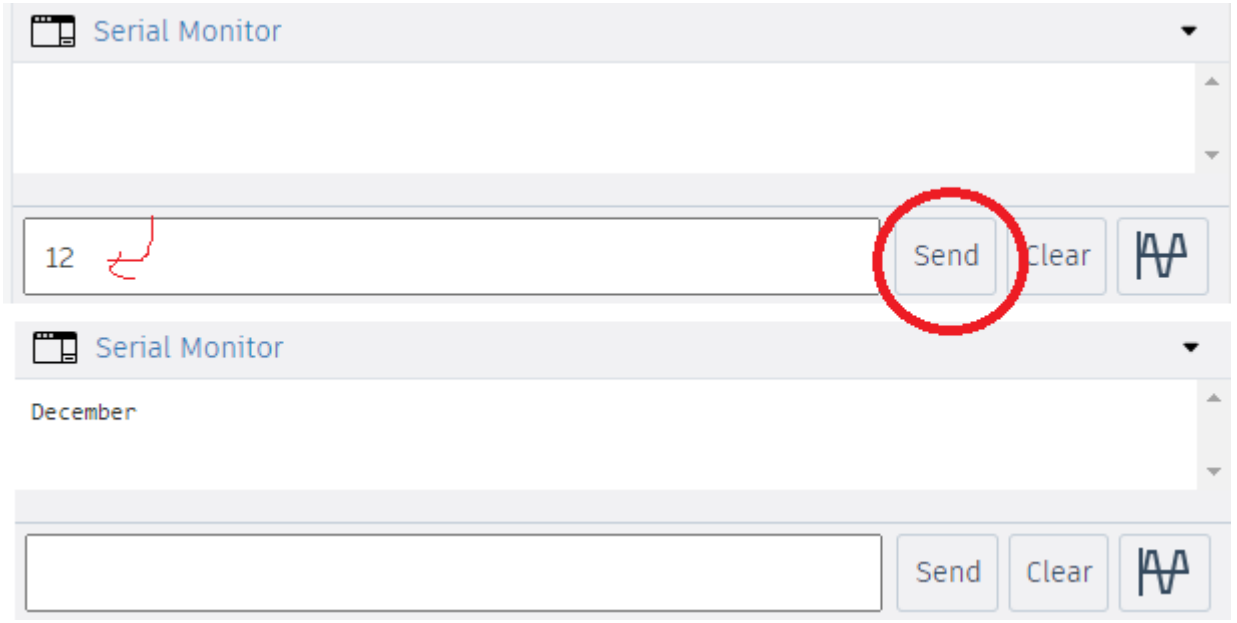
Conditional Examples #3

```
char *month_string(int m);

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

void loop()
{
  if (Serial.available()) {
    int x = Serial.parseInt();
    Serial.println(month_string(x));
  }
}
```

```
char *month_string(int m) {
  switch (m) {
    case 1: return "January";
    case 2: return "February";
    case 3: return "March";
    case 4: return "April";
    case 5: return "May";
    case 6: return "June";
    case 7: return "July";
    case 8: return "August";
    case 9: return "September";
    case 10: return "October";
    case 11: return "November";
    case 12: return "December";
    default: return "ERROR";
  }
}
```



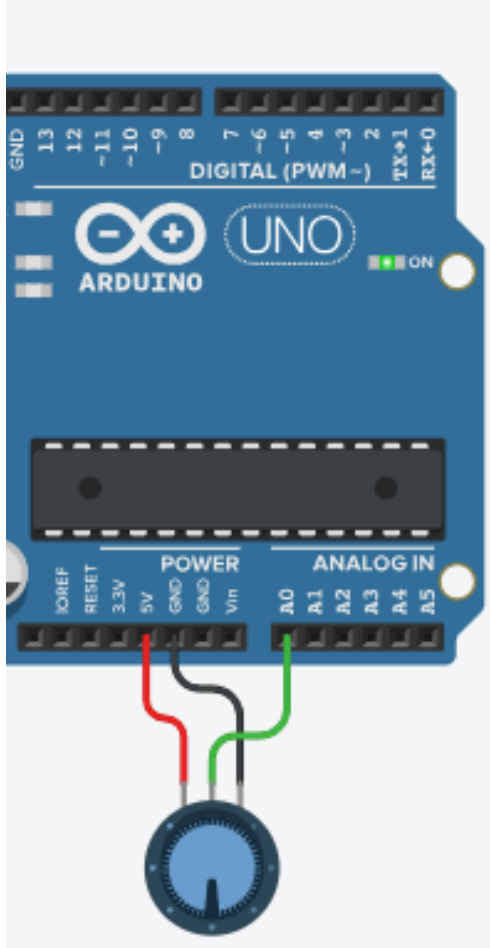
Conditional Examples #4

```
#define POT A0
#define LOW_LIMIT 2.25
#define HIGH_LIMIT 3.25

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

```
void loop()
{
  unsigned int potValue = analogRead(POT);
  float voltValue = potValue * 5.0 / 1023.0;
  if (voltValue <= LOW_LIMIT)
    Serial.println("LOWER THAN THE LOWER LIMIT");
  else if (voltValue >= HIGH_LIMIT)
    Serial.println("HIGHER THAN THE UPPER LIMIT");
  else
    Serial.println("LEVEL IS GOOD :)");
  delay(1000);
}
```

Conditional Examples #4 (cont'd)



Serial Monitor

```
LOWER THAN THE LOWER LIMIT
LOWER THAN THE LOWER LIMIT
LOWER THAN THE LOWER LIMIT
LOWER THAN THE LOWER LIMIT
LOWER THAN THE LOWER LIMIT
LOWER THAN THE LOWER LIMIT
LEVEL IS GOOD :)
LEVEL IS GOOD :)
HIGHER THAN THE UPPER LIMIT
HIGHER THAN THE UPPER LIMIT
HIGHER THAN THE UPPER LIMIT
LEVEL IS GOOD :)
LEVEL IS GOOD :)
LEVEL IS GOOD :)
LEVEL IS GOOD :)
LEVEL IS GOOD :)
LEVEL IS GOOD :)
```

Left to Students: Debug & Fix the problem

Please, fix the mistake hidden in the code given on the right ☺

```
#define MT_RATE (0.4)
#define FN_RATE (0.6)
unsigned char mtExam, fnExam, grade;
void setup() { Serial.begin(9600); }
void loop() {
  Serial.println("For a new grade calculation");
  Serial.print("Mid-Term Exam: ");
  mtExam = Serial.parseInt();
  Serial.print("Final Exam: ");
  fnExam = Serial.parseInt();
  grade = MT_RATE * mtExam + FN_RATE * fnExam;
  Serial.println("The grade is " + grade);
  delay(1000);
  Serial.println("\n---------\n");
}
```

Left to Students: Conditional Examples

- Write the following programs both in Arduino C and Rpi Pico CircuitPython and simulate them:

1. Find the minimum, maximum, median, and average of the last three numbers read from serial port and report them to the serial port back.
2. Read the user name from the serial port. If the user name equals to «admin», then request a pasword and compare it whether «1234» or not ☺. For other users, program sends a message like «Hello Yalcin!» (if the user name is «Yalcin») to the serial port.
3. There are 4 buttons and 1 LED. Buttons should be added with PULLUP resistors. LED can be BUILTIN_LED. When one of the first three buttons is pressed at least, then LED is ON. When the 4th button is pressed, then LED is OFF. After 3 seconds waiting, user can activate LED again by pressing one of the first three buttons. Status should be reported in every stiuation via the serial port.
4. There are 3 buttons and 1 LED. Buttons should be added with PULLUP resistors. LED can be BUILTIN_LED. When one of the buttons is pressed & released, then it toggles the status of the LED (i.e. if LED is ON then it becomes OFF and vice versa).

Thanks for listening ☺

YALÇIN İŞLER

Assoc. Prof.