Activity 4 : Programming Embedded Systems (Having Fun with Arduino)

Group No : 14
Group Member :

1. Thanach Silapatcharanun
2. Punnawat Honglerdnapakul
3. Nutthapat Pongtanyavichai
4. Vijak Khajornritdacha

A Chance to be “Outstanding”

Prerequisite: You have to complete all parts before doing this part.

Write Arduino code to decode “Morse Code”. The input of the “Morse Code” is the push button. As shown in the figure (in page 2), a dot is represented by a short push, a dash is represented by a long push. You may determine short and long push by yourself.

There are 3 types of command for the serial input

- ‘s’ to start input “Morse Code” for a character.
- ‘n’ to stop input “Morse Code” of a character. Also, output the character to the serial output and replay the “Morse Code” of that character with LED.
- ‘r’ to show the result of all the characters you entered.

After you have finished, call a TA or an instructor to check your output. Also, put the code in the answer box.

Example Video Clip : https://youtu.be/rD7mwFlDQbQ

```
int ledPin = 2;
int buttonPin = 3;
int holdDuration = 0;
int LONG_HOLD = 1000;
int prevButtonStatus = 0;
int isInputting = 0;

String morseBuffer = String();
```

Activity 4: Programming Embedded Systems (Outstanding)

```
String morseText = String();
char outputBuffer[500] = {};

void setup()
{
    pinMode(ledPin, OUTPUT);
    pinMode(buttonPin, INPUT);
    Serial.begin(9600);
}

void loop()
{
    int buttonStatus = digitalRead(buttonPin);
    digitalWrite(ledPin, buttonStatus);

    // Read serial input
    if (Serial.available() > 0)
    {
        char inChar = Serial.read();
        if (inChar == 's')
        {
            isInputting = 1;
            Serial.println("Start Recording");
        }
        else if (inChar == 'n')
        {

            isInputting = 0;
            Serial.println(morseBuffer);
            char morseCode = decodeMorse();

            lightLed();
            Serial.println("Stop Recording");
            Serial.println(morseCode);
```

Activity 4: Programming Embedded Systems (Outstanding)

```
            morseText += morseCode;
            morseBuffer = String();
        }
        else if (inChar == 'r')
        {
            Serial.print("result: ");
            Serial.println(morseText);
            morseText = String();
        }
    }

    if (isInputting)
        solve(buttonStatus);
}

void solve(int buttonStatus)
{
    if (buttonStatus == 1)
    {
        holdDuration++;
    }

    if (isInputting && prevButtonStatus != buttonStatus &&
buttonStatus == 0)
    {
        if (holdDuration > LONG_HOLD)
        {
            Serial.println("-");
            morseBuffer += '-';
        }
        else
        {
            Serial.println(".");
            morseBuffer += '.';
```

Activity 4: Programming Embedded Systems (Outstanding)

```
        }

        // sprintf(outputBuffer, "%d")
        Serial.println(holdDuration);
        holdDuration = 0;
    }

    prevButtonStatus = buttonStatus;
}

void lightLed()
{
    digitalWrite(ledPin, LOW);
    delay(500);
    for (int i = 0; i < morseBuffer.length(); i++)
    {
        char c = morseBuffer.charAt(i);
        int duration = 100;
        if (c == '-')
        {
            duration = 1000;
        }
        digitalWrite(ledPin, HIGH);
        delay(duration);
        digitalWrite(ledPin, LOW);
        delay(500);
    }
}

char decodeMorse()
{
    if (morseBuffer.compareTo(String(".-")) == 0)
        return 'A';
    else if (morseBuffer.compareTo(String("-...")) == 0)
```

Activity 4: Programming Embedded Systems (Outstanding)

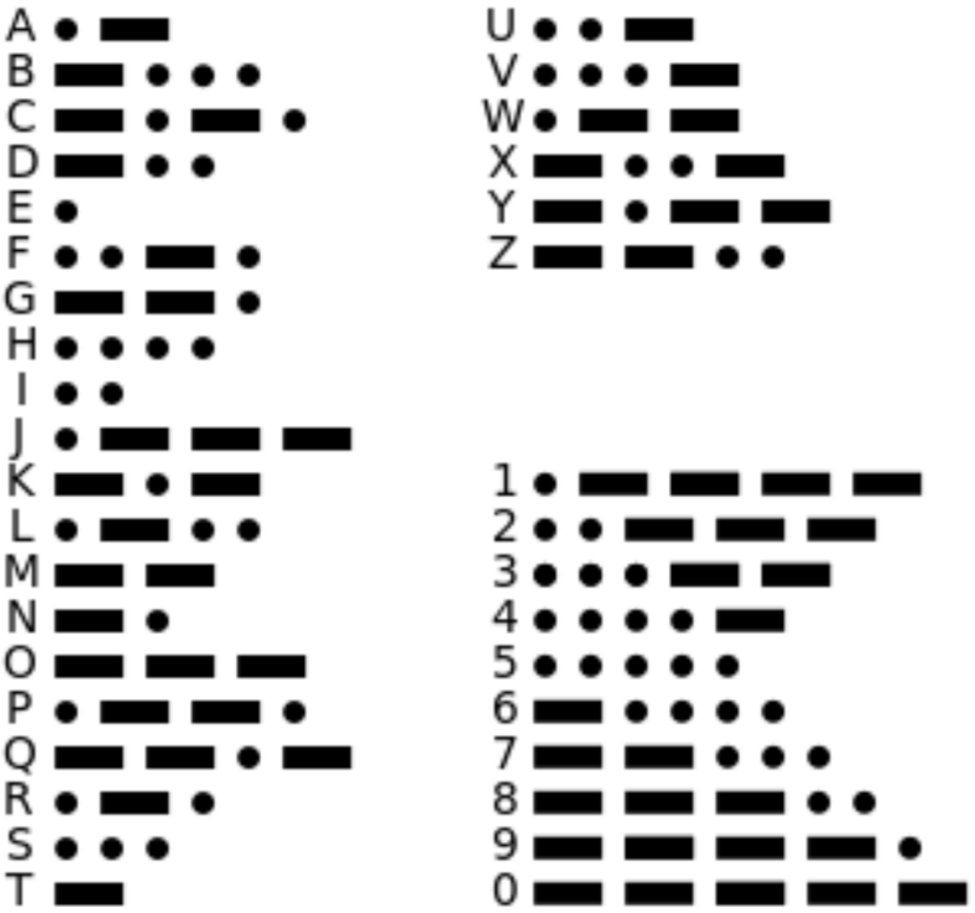
```
        return 'B';
    else if (morseBuffer.compareTo(String("-.-.")) == 0)
        return 'C';
    else if (morseBuffer.compareTo(String("-..")) == 0)
        return 'D';
    else if (morseBuffer.compareTo(String(".")) == 0)
        return 'E';
    else if (morseBuffer.compareTo(String("..-.")) == 0)
        return 'F';
    else if (morseBuffer.compareTo(String("--.")) == 0)
        return 'G';
    else if (morseBuffer.compareTo(String("....")) == 0)
        return 'H';
    else if (morseBuffer.compareTo(String("..")) == 0)
        return 'I';
    else if (morseBuffer.compareTo(String(".---")) == 0)
        return 'J';
    else if (morseBuffer.compareTo(String("-.-")) == 0)
        return 'K';
    else if (morseBuffer.compareTo(String(".-..")) == 0)
        return 'L';
    else if (morseBuffer.compareTo(String("--")) == 0)
        return 'M';
    else if (morseBuffer.compareTo(String("-.")) == 0)
        return 'N';
    else if (morseBuffer.compareTo(String("---")) == 0)
        return 'O';
    else if (morseBuffer.compareTo(String(".--.")) == 0)
        return 'P';
    else if (morseBuffer.compareTo(String("--.-")) == 0)
        return 'Q';
    else if (morseBuffer.compareTo(String(".-.")) == 0)
        return 'R';
    else if (morseBuffer.compareTo(String("...")) == 0)
```

Activity 4: Programming Embedded Systems (Outstanding)

```
        return 'S';
    else if (morseBuffer.compareTo(String("-")) == 0)
        return 'T';
    else if (morseBuffer.compareTo(String("..-")) == 0)
        return 'U';
    else if (morseBuffer.compareTo(String("...-")) == 0)
        return 'V';
    else if (morseBuffer.compareTo(String(".--")) == 0)
        return 'W';
    else if (morseBuffer.compareTo(String("-..-")) == 0)
        return 'X';
    else if (morseBuffer.compareTo(String("-.--")) == 0)
        return 'Y';
    else if (morseBuffer.compareTo(String("--..")) == 0)
        return 'Z';
    else if (morseBuffer.compareTo(String("-----")) == 0)
        return '0';
    else if (morseBuffer.compareTo(String(".----")) == 0)
        return '1';
    else if (morseBuffer.compareTo(String("..---")) == 0)
        return '2';
    else if (morseBuffer.compareTo(String("...--")) == 0)
        return '3';
    else if (morseBuffer.compareTo(String("....-")) == 0)
        return '4';
    else if (morseBuffer.compareTo(String(".....")) == 0)
        return '5';
    else if (morseBuffer.compareTo(String("-....")) == 0)
        return '6';
    else if (morseBuffer.compareTo(String("--...")) == 0)
        return '7';
    else if (morseBuffer.compareTo(String("---..")) == 0)
        return '8';
    else if (morseBuffer.compareTo(String("----.")) == 0)
```

Activity 4: Programming Embedded Systems (Outstanding)

```
        return '9';
}
```



— THIS IS THE END OF OUTSTANDING PART —