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

Group No : 35
Group Member :

1. Chotpisit Adunsehawat 6531313221
2. Sarin Yongsuwan 6532182521
3. Punyaphat Surakiatkamjorn 6432106821
4. Athiwat Kongkaeo 6531347621

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

```
const int ledpin = 2;
const int button = 3;

const int shortTime = 4;
const int endTime = 8;
int status = 0;
int buttonState = 0;


int codes[10] = {0,0,0,0,0,0,0,0,0,0};
int rangeBar[10] = {0,0,0,0,0,0,0,0,0,0};
int waitingTime[10] = {0,0,0,0,0,0,0,0,0,0};
```

Activity 4: Programming Embedded Systems (Outstanding)

```
int currentIndex = 0;
bool isDecoding = 0;
bool isPressed = 0;
int rangeCount = 0;
String res = "";

void setup()
{
  Serial.begin(115200);
  Serial.println("start program");
  pinMode(ledpin, OUTPUT);
  pinMode(button, INPUT);
}

void loop(){
  char inChar = Serial.read();
  if(inChar == 's'){
    for(int i=0;i<10;++i){
      codes[i] = 0;
            rangeBar[i] = 0;
            waitingTime[i] = 0;
    }
      status = 1;
    currentIndex = 0;
    Serial.println("s");
  }
  if(inChar == 'n'){
    Serial.println("n");
      replay();
    status = 0;
    currentIndex = 0;
  }
  if(inChar == 'r'){
    Serial.println("r");
    res += decodeToChar();
    Serial.println(res);
  }
  if(status == 1){
      getCode();
  }
}

void replay(){
  delay(500);
  digitalWrite(ledpin, LOW);
  for(int i=0;i<10;++i){
    for(int j=0;j<rangeBar[i];++j){
      digitalWrite(ledpin, HIGH);
      delay(1);
    }
    digitalWrite(ledpin, LOW);
```

Activity 4: Programming Embedded Systems (Outstanding)

```
    for(int j=0;j<waitingTime[i];++j){
      delay(1);
    }
  }
}

void getCode(){
  buttonState = digitalRead(button);

  // stat
  if(isPressed == 0 && buttonState == 1){
      if(currentIndex != 0){
      waitingTime[currentIndex-1] = rangeCount;
    }
    rangeCount = 0;
    isPressed = 1;
  }
  // end
  if(isPressed == 1 && buttonState == 0){
    if(rangeCount > 250){
      codes[currentIndex] = 2;
    }else{
      codes[currentIndex] = 1;
    }
    rangeBar[currentIndex] = rangeCount;
    rangeCount = 0;
    currentIndex += 1;
    isPressed = 0;
  }
  rangeCount+=1;

  if(buttonState == 1){
    digitalWrite(ledpin, HIGH);
  }
  else{
    digitalWrite(ledpin, LOW);
  }
  delay(1);
}

char decodeToChar()
{
  // A
  if ( codes[0] == 1 && codes[1] == 2 && codes[2] == 0 )
  {
    return 'A';
  }

  // B
  if ( codes[0] == 2 && codes[1] == 1 && codes[2] == 1 && codes[3] == 1
&& codes[4] == 0)
```

Activity 4: Programming Embedded Systems (Outstanding)

```
  {
    return 'B';
  }

  // C
  if ( codes[0] == 2 && codes[1] == 1 && codes[2] == 2 && codes[3] == 1
&& codes[4] == 0)
  {
    return 'C';
  }

  // D
  if ( codes[0] == 2 && codes[1] == 1 && codes[2] == 1 && codes[3] == 0 )
  {
    return 'D';
  }

  // E
  if ( codes[0] == 1 && codes[1] == 0 )
  {
    return 'E';
  }

  // F
  if ( codes[0] == 1 && codes[1] == 1 && codes[2] == 2 && codes[3] == 1
&& codes[4] == 0)
  {
    return 'F';
  }

  // G
  if ( codes[0] == 2 && codes[1] == 2 && codes[2] == 1 && codes[3] == 0)
  {
    return 'G';
  }

  // H
  if ( codes[0] == 1 && codes[1] == 1 && codes[2] == 1 && codes[3] == 1
&& codes[4] == 0)
  {
    return 'H';
  }

  // I
  if ( codes[0] == 1 && codes[1] == 1 && codes[2] == 0 )
  {
    return 'I';
  }

  // J
```

Activity 4: Programming Embedded Systems (Outstanding)

```
  if ( codes[0] == 1 && codes[1] == 2 && codes[2] == 2 && codes[3] == 2
&& codes[4] == 0)
  {
    return 'J';
  }

  // K
  if ( codes[0] == 2 && codes[1] == 1 && codes[2] == 2 && codes[3] == 0 )
  {
    return 'K';
  }

  // L
  if ( codes[0] == 1 && codes[1] == 2 && codes[2] == 1 && codes[3] == 1
&& codes[4] == 0)
  {
    return 'L';
  }

  // M
  if ( codes[0] == 2 && codes[1] == 2 && codes[2] == 0 )
  {
    return 'M';
  }

  // N
  if ( codes[0] == 2 && codes[1] == 1 && codes[2] == 0)
  {
    return 'N';
  }

  // O
  if ( codes[0] == 2 && codes[1] == 2 && codes[2] == 2 && codes[3] == 0)
  {
    return 'O';
  }

  // P
  if ( codes[0] == 1 && codes[1] == 2 && codes[2] == 2 && codes[3] == 1
&& codes[4] == 0)
  {
    return 'P';
  }

  // Q
  if ( codes[0] == 2 && codes[1] == 2 && codes[2] == 1 && codes[3] == 2
&& codes[4] == 0)
  {
    return 'Q';
  }
```

Activity 4: Programming Embedded Systems (Outstanding)

```
  // R
  if ( codes[0] == 1 && codes[1] == 2 && codes[2] == 1 && codes[3] == 0)
  {
    return 'R';
  }

  // S
  if ( codes[0] == 1 && codes[1] == 1 && codes[2] == 1 && codes[3] == 0)
  {
    return 'S';
  }

  // T
  if ( codes[0] == 2 && codes[1] == 0 )
  {
    return 'T';
  }

  // U
  if ( codes[0] == 1 && codes[1] == 1 && codes[2] == 2 && codes[3] == 0)
  {
    return 'U';
  }

  // V
  if ( codes[0] == 1 && codes[1] == 1 && codes[2] == 1 && codes[3] == 2
&& codes[4] == 0)
  {
    return 'V';
  }

  // W
  if ( codes[0] == 1 && codes[1] == 2 && codes[2] == 2 && codes[3] == 0)
  {
    return 'W';
  }

  // X
  if ( codes[0] == 2 && codes[1] == 1 && codes[2] == 1 && codes[3] == 2
&& codes[4] == 0)
  {
    return 'X';
  }

  // Y
  if ( codes[0] == 2 && codes[1] == 1 && codes[2] == 2 && codes[3] == 2
&& codes[4] == 0)
  {
    return 'Y';
  }
```

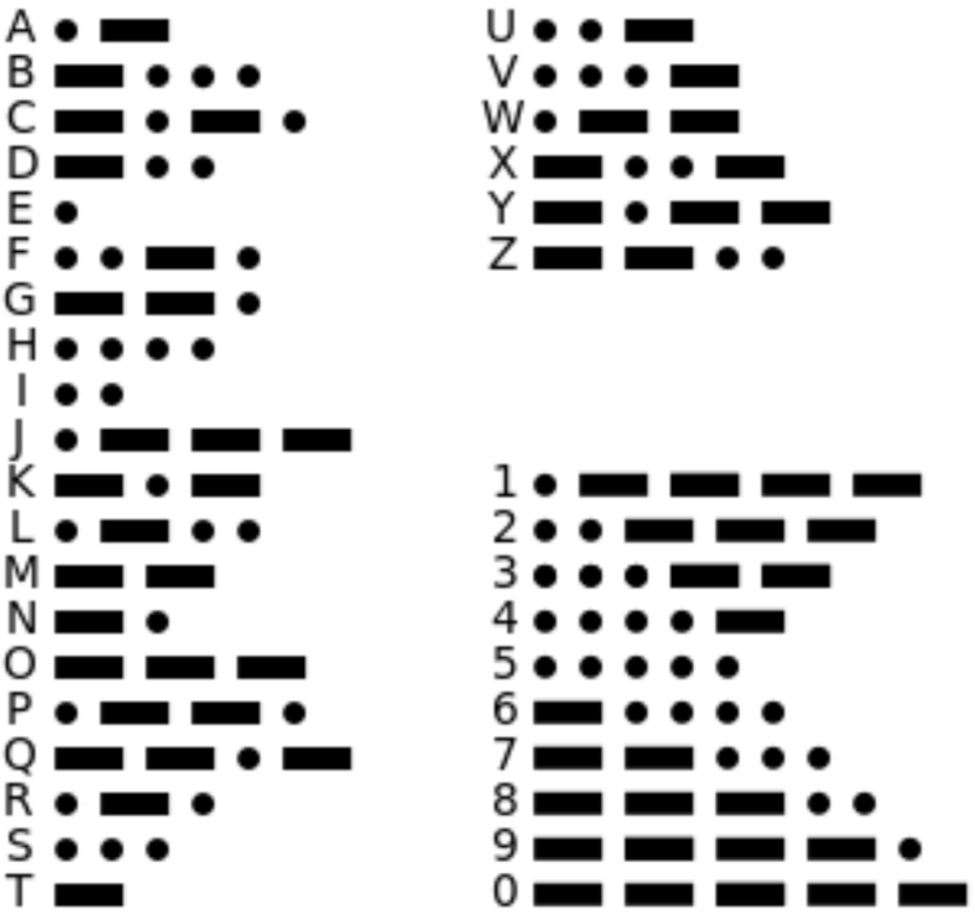
Activity 4: Programming Embedded Systems (Outstanding)

```
  // Z
  if ( codes[0] == 2 && codes[1] == 2 && codes[2] == 1 && codes[3] == 1
&& codes[4] == 0)
  {
    return 'Z';
  }

  // 1
  if ( codes[0] == 1 && codes[1] == 2 && codes[2] == 2 && codes[3] == 2
&& codes[4] == 2 && codes[5] == 0)
  {
    return '1';
  }

  // 2
  if ( codes[0] == 1 && codes[1] == 1 && codes[2] == 2 && codes[3] == 2
&& codes[4] == 2 && codes[5] == 0)
  {
    return '2';
  }

  // 3
  if ( codes[0] == 1 && codes[1] == 1 && codes[2] == 1 && codes[3] == 2
&& codes[4] == 2 && codes[5] == 0)
  {
    return '3';
  }

  // 4
  if ( codes[0] == 1 && codes[1] == 1 && codes[2] == 1 && codes[3] == 1
&& codes[4] == 2 && codes[5] == 0)
  {
    return '4';
  }

  // 5
  if ( codes[0] == 1 && codes[1] == 1 && codes[2] == 1 && codes[3] == 1
&& codes[4] == 1 && codes[5] == 0)
  {
    return '5';
  }

  // 6
  if ( codes[0] == 2 && codes[1] == 1 && codes[2] == 1 && codes[3] == 1
&& codes[4] == 1 && codes[5] == 0)
  {
    return '6';
  }

  // 7
```

Activity 4: Programming Embedded Systems (Outstanding)

```
  if ( codes[0] == 2 && codes[1] == 2 && codes[2] == 1 && codes[3] == 1
&& codes[4] == 1 && codes[5] == 0)
  {
    return '7';
  }

  // 8
  if ( codes[0] == 2 && codes[1] == 2 && codes[2] == 2 && codes[3] == 1
&& codes[4] == 1 && codes[5] == 0)
  {
    return '8';
  }

  // 9
  if ( codes[0] == 2 && codes[1] == 2 && codes[2] == 2 && codes[3] == 2
&& codes[4] == 1 && codes[5] == 0)
  {
    return '9';
  }

  // 0
  if ( codes[0] == 2 && codes[1] == 2 && codes[2] == 2 && codes[3] == 2
&& codes[4] == 2 && codes[5] == 0)
  {
    return '0';
  }


  return ' ';
}
```

Activity 4: Programming Embedded Systems (Outstanding)



— THIS IS THE END OF OUTSTANDING PART —