

```

//4BI6 - Group 13
//Jonathan Hernandez
//Braille Teaching Device - Microcontroller Code

//Global variable. Determines if incoming letter
//is lowercase or contains capital/number.

//0 = lowercase
//1 = Capital
//2 = Number
int capFunc = 0;

//char Array to store incoming text
char words[4];
char stored[6];

//var to check if data was sent
int dataSent = 0;

volatile int state = LOW;

void inter1()
{
    //Button #1 is pushed.
    if(state == LOW)
    {
        Serial.write("a");
        state = !state;
    }
}

void inter2()
{
    //Button #2 is pushed.
    if(state == LOW)
    {
        Serial.write("b");
        state = !state;
    }
}

ISR( PCINT2_vect )
{
    //Button #3 or 4 is pushed. Determine which.
    if(state == LOW)
    {
        if(digitalRead(4) == HIGH) // evaluates to TRUE when PortD bit #4 (Arduino
digital pin #5) is HIGH
        {
            //Button #3 is pushed.
            Serial.write("c");
            state = !state;
        }
        else if(digitalRead(5) == HIGH) // evaluates to TRUE when PortD bit #5
(Arduino digital pin #5) is HIGH
        {
            //Button #4 is pushed.
            Serial.write("d");
            state = !state;
        }
    }
}

```

```

    }
  }
}

void setup()
{
  interrupts();
  /*Code used to initiate DLATCH Pins*/
  pinMode(6,OUTPUT);
  digitalWrite(6,HIGH);
  pinMode(7,OUTPUT);
  digitalWrite(7,HIGH);

  pinMode(8, OUTPUT);
  digitalWrite(8,HIGH);
  pinMode(9, OUTPUT);
  digitalWrite(9,HIGH);
  pinMode(10, OUTPUT);
  digitalWrite(10,HIGH);
  pinMode(11, OUTPUT);
  digitalWrite(11,HIGH);
  pinMode(12, OUTPUT);
  digitalWrite(12,HIGH);
  pinMode(13, OUTPUT);
  digitalWrite(13,HIGH);

  //Initially writes to first Braille Letter
  pinMode(0, OUTPUT);
  digitalWrite(0,LOW);
  pinMode(1, OUTPUT);
  digitalWrite(1,LOW);

  delay(100);
  //Now write to second Braille letter
  digitalWrite(1,HIGH);

  delay(100);
  //Now write to third Braille letter
  digitalWrite(0, HIGH);
  digitalWrite(1, LOW);

  delay(100);
  //Now write to fourth Braille letter
  digitalWrite(1, HIGH);

  delay(100);
  //Set back to fourth letter
  digitalWrite(0,LOW);
  digitalWrite(1,LOW);

  //Set digital Pins for Buttons/Interrupts
  pinMode(4, INPUT);
  pinMode(5, INPUT);

  attachInterrupt(0, inter1, RISING);
  attachInterrupt(1, inter2, RISING);

```

```

//digital Pin 4
PCICR |= (1 << PCIE2);
PCMSK2 |= (1 << PCINT20);
PCMSK2 |= (1 << PCINT21);

//digital Pin 5
//PCICR |= (1 << PCIE1);
//PCMSK1 |= (1 << PCINT21);

Serial.begin(9600);
}

void loop()
{
    if(state ==HIGH)
    {
        state = LOW;
        delay(600);
    }

    char var;
    int count =0;
    int index = 0;
    int index2 = 0;
    if(Serial.available() > 0 && dataSent == 0)
    {
        //Clear words[] data.
        words[0] = ' ';
        words[1] = ' ';
        words[2] = ' ';
        words[3] = ' ';

        stored[0] = ' ';
        stored[1] = ' ';
        stored[2] = ' ';
        stored[3] = ' ';
        stored[4] = ' ';
        stored[5] = ' ';

        char inLetter= ' '; // Where to store the character read
        char dumpLetter= ' '; // Used to dump excess letters.
        index2 = 0; // Index into array; where to store the character

        while(Serial.available() > 0) // Don't read unless
        // there you know there is data
        {

            delay(200);
            if(index2 <6) // One less than the size of the array
            {
                inLetter = Serial.read(); // Read a character
                stored[index2] = inLetter; // Store it
                index2++; // Increment where to write next
            }

```

```

        else
        {
            index2 = 6;
            dumpLetter = Serial.read();
        }
    }

    if(stored[0] == '0')
    {
        capFunc = 0;
    }
    else if(stored[0] == '1')
    {
        capFunc = 1;
    }
    else if(stored[0] == '2')
    {
        capFunc = 2;
    }
    else if(stored[0] == '3')
    {
        capFunc = 3;
    }

    index = 0;
    if(stored[1] == '1')
    {
        index = 1;
    }
    else if(stored[1] == '2')
    {
        index = 2;
    }
    else if(stored[1] == '3')
    {
        index = 3;
    }
    else if(stored[1] == '4')
    {
        index = 4;
    }

    count=2;
    while(stored[count] != ' ' && (count-2) < index)
    {
        words[count-2] = stored[count];
        count++;
    }

    //Call necessary type of function.
    //Send # of characters used in English string.
    if(capFunc == 0)
    {
        regWord(index);
    }
    else if (capFunc == 1)
    {

```

```

        capSet(index);
        regWord(index);
    }
    else if (capFunc == 2)
    {
        numSet(index);
        regWord(index);
    }
    else if (capFunc == 3)
    { //Used if word is punctuation
        regPunc(index);
    }

    Serial.write(words[0]);
}
}

```

//Set the Braille letters. Start with decoder #4.

```

void regWord(int index)
{
    int currInd = index-1;
    int count = 0;
    while(count < index)
    {
        if(count == 0)
        {
            digitalWrite(0,HIGH);
            digitalWrite(1,HIGH);
        }
        else if (count == 1)
        {
            digitalWrite(0,HIGH);
            digitalWrite(1,LOW);
        }
        else if(count == 2)
        {
            digitalWrite(0,LOW);
            digitalWrite(1,HIGH);
        }
        else if(count ==3)
        {
            digitalWrite(0,LOW);
            digitalWrite(1,LOW);
        }
    }
}

```

```

//Set the proper letters
setBraille(words[currInd]);

```

```

//Increment count and move to next letter.
count++;
currInd = index -1 -count;
}
}

```

```

void regPunc(int index)
{
    int currInd = index-1;
}

```

```

int count = 0;
while(count < index)
{
    if(count == 0)
    {
        digitalWrite(0,HIGH);
        digitalWrite(1,HIGH);
    }
    else if (count == 1)
    {
        digitalWrite(0,HIGH);
        digitalWrite(1,LOW);
    }
    else if(count == 2)
    {
        digitalWrite(0,LOW);
        digitalWrite(1,HIGH);
    }
    else if(count ==3)
    {
        digitalWrite(0,LOW);
        digitalWrite(1,LOW);
    }
    }

    //Set the proper letters
    setPunc(words[currInd]);

    //Increment count and move to next letter.
    delay(100);
    count++;
    currInd = index -1 -count;
}
}

void capSet(int index)
{
    int count = 0;
    char temp[4];
    while(count < index+1)
    {
        if(count == 0)
        {
            temp[0] = '+';
        }
        else if(words[count-1] == 'A')
        {
            temp[count] = 'a';
        }
        else if(words[count-1] == 'B')
        {
            temp[count] = 'b';
        }
        else if(words[count-1] == 'C')
        {
            temp[count] = 'c';
        }
        else if(words[count-1] == 'D')

```

```
{
    temp[count] = 'd';
}
else if(words[count-1] == 'E')
{
    temp[count] = 'e';
}
else if(words[count-1] == 'F')
{
    temp[count] = 'f';
}
else if(words[count-1] == 'G')
{
    temp[count] = 'g';
}
else if(words[count-1] == 'H')
{
    temp[count] = 'h';
}
else if(words[count-1] == 'I')
{
    temp[count] = 'i';
}
else if(words[count-1] == 'J')
{
    temp[count] = 'j';
}
else if(words[count-1] == 'K')
{
    temp[count] = 'k';
}
else if(words[count-1] == 'L')
{
    temp[count] = 'l';
}
else if(words[count-1] == 'M')
{
    temp[count] = 'm';
}
else if(words[count-1] == 'N')
{
    temp[count] = 'n';
}
else if(words[count-1] == 'O')
{
    temp[count] = 'o';
}
else if(words[count-1] == 'P')
{
    temp[count] = 'p';
}
else if(words[count-1] == 'Q')
{
    temp[count] = 'q';
}
else if(words[count-1] == 'R')
{

```

```

        temp[count] = 'r';
    }
    else if(words[count-1] == 'S')
    {
        temp[count] = 's';
    }
    else if(words[count-1] == 'T')
    {
        temp[count] = 't';
    }
    else if(words[count-1] == 'U')
    {
        temp[count] = 'u';
    }
    else if(words[count-1] == 'V')
    {
        temp[count] = 'v';
    }
    else if(words[count-1] == 'W')
    {
        temp[count] = 'w';
    }
    else if(words[count-1] == 'X')
    {
        temp[count] = 'x';
    }
    else if(words[count-1] == 'Y')
    {
        temp[count] = 'y';
    }
    else if(words[count-1] == 'Z')
    {
        temp[count] = 'z';
    }
    else
    {
        temp[count] = words[count-1];
    }
    count++;
}

for(int i = 0; i < numSet(index))
{
    int count = 0;
    char temp[4];
    while(count < index+1)
    {
        if(count == 0)
        {
            temp[0] = '#';
        }
        else if(words[count-1] == '1' && count != 0)
        {
            temp[count] = 'a';
        }
        else if(words[count-1] == '2' && count != 0)
        {

```

```

        temp[count] = 'b';
    }
    else if(words[count-1] == '3' && count != 0)
    {
        temp[count] = 'c';
    }
    else if(words[count-1] == '4' && count != 0)
    {
        temp[count] = 'd';
    }
    else if(words[count-1] == '5' && count != 0)
    {
        temp[count] = 'e';
    }
    else if(words[count-1] == '6' && count != 0)
    {
        temp[count] = 'f';
    }
    else if(words[count-1] == '7' && count != 0)
    {
        temp[count] = 'g';
    }
    else if(words[count-1] == '8' && count != 0)
    {
        temp[count] = 'h';
    }
    else if(words[count-1] == '9' && count != 0)
    {
        temp[count] = 'i';
    }
    else if(words[count-1] == '0' && count != 0)
    {
        temp[count] = 'j';
    }
    count++;
}

for(int i = 0; i<(index+1); i++)
{
    words[i] = temp[i];
}
}

void setBraille(char var)
{
    if(var == '+')
    {
        digitalWrite(13,HIGH);
        digitalWrite(12,HIGH);
        digitalWrite(11,HIGH);
        digitalWrite(10,HIGH);
        digitalWrite(9,HIGH);
        digitalWrite(8,LOW);
    }
    else if(var == '#')
    {
        digitalWrite(13,HIGH);
    }
}

```

```
    digitalWrite(12,HIGH);
    digitalWrite(11,LOW);
    digitalWrite(10,LOW);
    digitalWrite(9,LOW);
    digitalWrite(8,LOW);
}
else if(var == 'a')
{
    digitalWrite(13,LOW);
    digitalWrite(12,HIGH);
    digitalWrite(11,HIGH);
    digitalWrite(10,HIGH);
    digitalWrite(9,HIGH);
    digitalWrite(8,HIGH);
}
else if(var == 'b')
{
    digitalWrite(13,LOW);
    digitalWrite(12,LOW);
    digitalWrite(11,HIGH);
    digitalWrite(10,HIGH);
    digitalWrite(9,HIGH);
    digitalWrite(8,HIGH);
}
else if(var == 'c')
{
    digitalWrite(13,LOW);
    digitalWrite(12,HIGH);
    digitalWrite(11,HIGH);
    digitalWrite(10,LOW);
    digitalWrite(9,HIGH);
    digitalWrite(8,HIGH);
}
else if(var == 'd')
{
    digitalWrite(13,LOW);
    digitalWrite(12,HIGH);
    digitalWrite(11,HIGH);
    digitalWrite(10,LOW);
    digitalWrite(9,LOW);
    digitalWrite(8,HIGH);
}
else if(var == 'e')
{
    digitalWrite(13,LOW);
    digitalWrite(12,HIGH);
    digitalWrite(11,HIGH);
    digitalWrite(10,HIGH);
    digitalWrite(9,LOW);
    digitalWrite(8,HIGH);
}
else if(var == 'f')
{
    digitalWrite(13,LOW);
    digitalWrite(12,LOW);
    digitalWrite(11,HIGH);
    digitalWrite(10,LOW);
}
```

```

        digitalWrite(9,HIGH);
        digitalWrite(8,HIGH);
    }
    else if(var == 'g')
    {
        digitalWrite(13,LOW);
        digitalWrite(12,LOW);
        digitalWrite(11,HIGH);
        digitalWrite(10,LOW);
        digitalWrite(9,LOW);
        digitalWrite(8,HIGH);
    }
    else if(var == 'h')
    {
        digitalWrite(13,LOW);
        digitalWrite(12,LOW);
        digitalWrite(11,HIGH);
        digitalWrite(10,HIGH);
        digitalWrite(9,LOW);
        digitalWrite(8,HIGH);
    }
    else if(var == 'i')
    {
        digitalWrite(13,HIGH);
        digitalWrite(12,LOW);
        digitalWrite(11,HIGH);
        digitalWrite(10,LOW);
        digitalWrite(9,HIGH);
        digitalWrite(8,HIGH);
    }
    else if(var == 'j')
    {
        digitalWrite(13,HIGH);
        digitalWrite(12,LOW);
        digitalWrite(11,HIGH);
        digitalWrite(10,LOW);
        digitalWrite(9,LOW);
        digitalWrite(8,HIGH);
    }
    else if(var == 'k')
    {
        digitalWrite(13,LOW);
        digitalWrite(12,HIGH);
        digitalWrite(11,LOW);
        digitalWrite(10,HIGH);
        digitalWrite(9,HIGH);
        digitalWrite(8,HIGH);
    }
    else if(var == 'l')
    {
        digitalWrite(13,LOW);
        digitalWrite(12,LOW);
        digitalWrite(11,LOW);
        digitalWrite(10,HIGH);
        digitalWrite(9,HIGH);
        digitalWrite(8,HIGH);
    }
}

```

```
else if(var == 'm')
{
    digitalWrite(13,LOW);
    digitalWrite(12,HIGH);
    digitalWrite(11,LOW);
    digitalWrite(10,LOW);
    digitalWrite(9,HIGH);
    digitalWrite(8,HIGH);
}
else if(var == 'n')
{
    digitalWrite(13,LOW);
    digitalWrite(12,HIGH);
    digitalWrite(11,LOW);
    digitalWrite(10,LOW);
    digitalWrite(9,LOW);
    digitalWrite(8,HIGH);
}
else if(var == 'o')
{
    digitalWrite(13,LOW);
    digitalWrite(12,HIGH);
    digitalWrite(11,LOW);
    digitalWrite(10,HIGH);
    digitalWrite(9,LOW);
    digitalWrite(8,HIGH);
}
else if(var == 'p')
{
    digitalWrite(13,LOW);
    digitalWrite(12,LOW);
    digitalWrite(11,LOW);
    digitalWrite(10,LOW);
    digitalWrite(9,HIGH);
    digitalWrite(8,HIGH);
}
else if(var == 'q')
{
    digitalWrite(13,LOW);
    digitalWrite(12,LOW);
    digitalWrite(11,LOW);
    digitalWrite(10,LOW);
    digitalWrite(9,LOW);
    digitalWrite(8,HIGH);
}
else if(var == 'r')
{
    digitalWrite(13,LOW);
    digitalWrite(12,LOW);
    digitalWrite(11,LOW);
    digitalWrite(10,HIGH);
    digitalWrite(9,LOW);
    digitalWrite(8,HIGH);
}
else if(var == 's')
{
    digitalWrite(13,HIGH);
```

```
    digitalWrite(12,LOW);
    digitalWrite(11,LOW);
    digitalWrite(10,LOW);
    digitalWrite(9,HIGH);
    digitalWrite(8,HIGH);
}
else if(var == 't')
{
    digitalWrite(13,HIGH);
    digitalWrite(12,LOW);
    digitalWrite(11,LOW);
    digitalWrite(10,LOW);
    digitalWrite(9,LOW);
    digitalWrite(8,HIGH);
}
else if(var == 'u')
{
    digitalWrite(13,LOW);
    digitalWrite(12,HIGH);
    digitalWrite(11,LOW);
    digitalWrite(10,HIGH);
    digitalWrite(9,HIGH);
    digitalWrite(8,LOW);
}
else if(var == 'v')
{
    digitalWrite(13,LOW);
    digitalWrite(12,LOW);
    digitalWrite(11,LOW);
    digitalWrite(10,HIGH);
    digitalWrite(9,HIGH);
    digitalWrite(8,LOW);
}
else if(var == 'w')
{
    digitalWrite(13,HIGH);
    digitalWrite(12,LOW);
    digitalWrite(11,HIGH);
    digitalWrite(10,LOW);
    digitalWrite(9,LOW);
    digitalWrite(8,LOW);
}
else if(var == 'x')
{
    digitalWrite(13,LOW);
    digitalWrite(12,HIGH);
    digitalWrite(11,LOW);
    digitalWrite(10,LOW);
    digitalWrite(9,HIGH);
    digitalWrite(8,LOW);
}
else if(var == 'y')
{
    digitalWrite(13,LOW);
    digitalWrite(12,HIGH);
    digitalWrite(11,LOW);
    digitalWrite(10,LOW);
}
```

```

        digitalWrite(9,LOW);
        digitalWrite(8,LOW);
    }
    else if(var == 'z')
    {
        digitalWrite(13,LOW);
        digitalWrite(12,HIGH);
        digitalWrite(11,LOW);
        digitalWrite(10,HIGH);
        digitalWrite(9,LOW);
        digitalWrite(8,LOW);
    }
}

```

```

void setPunc(char var)
{
    if(var == 'a')
    { //Apostrophe
        digitalWrite(13,HIGH);
        digitalWrite(12,HIGH);
        digitalWrite(11,LOW);
        digitalWrite(10,HIGH);
        digitalWrite(9,HIGH);
        digitalWrite(8,HIGH);
    }
    else if(var == 'b' || var == 'c')
    { //Open and close brackets are same in Braille.
        digitalWrite(13,HIGH);
        digitalWrite(12,LOW);
        digitalWrite(11,LOW);
        digitalWrite(10,HIGH);
        digitalWrite(9,LOW);
        digitalWrite(8,LOW);
    }
    else if(var == 'd')
    { //Comma
        digitalWrite(13,HIGH);
        digitalWrite(12,LOW);
        digitalWrite(11,HIGH);
        digitalWrite(10,HIGH);
        digitalWrite(9,HIGH);
        digitalWrite(8,HIGH);
    }
    else if(var == 'e')
    { //Exclamation
        digitalWrite(13,HIGH);
        digitalWrite(12,LOW);
        digitalWrite(11,LOW);
        digitalWrite(10,HIGH);
        digitalWrite(9,LOW);
        digitalWrite(8,HIGH);
    }
    else if(var == 'f')
    { //Hyphen
        digitalWrite(13,HIGH);
        digitalWrite(12,HIGH);
    }
}

```

```

        digitalWrite(11,LOW);
        digitalWrite(10,HIGH);
        digitalWrite(9,HIGH);
        digitalWrite(8,LOW);
    }
    else if(var == 'g')
    { //Period
        digitalWrite(13,HIGH);
        digitalWrite(12,LOW);
        digitalWrite(11,HIGH);
        digitalWrite(10,HIGH);
        digitalWrite(9,LOW);
        digitalWrite(8,LOW);
    }
    else if(var == 'h')
    { //Question Mark
        digitalWrite(13,HIGH);
        digitalWrite(12,LOW);
        digitalWrite(11,LOW);
        digitalWrite(10,HIGH);
        digitalWrite(9,HIGH);
        digitalWrite(8,LOW);
    }
    else if(var == 'i')
    { //Open Quotations
        digitalWrite(13,HIGH);
        digitalWrite(12,LOW);
        digitalWrite(11,LOW);
        digitalWrite(10,HIGH);
        digitalWrite(9,HIGH);
        digitalWrite(8,LOW);
    }
    else if(var == 'j')
    { //Close Quotations
        digitalWrite(13,HIGH);
        digitalWrite(12,HIGH);
        digitalWrite(11,LOW);
        digitalWrite(10,HIGH);
        digitalWrite(9,LOW);
        digitalWrite(8,LOW);
    }
    else if(var == 'k')
    { //Semicolon
        digitalWrite(13,HIGH);
        digitalWrite(12,LOW);
        digitalWrite(11,LOW);
        digitalWrite(10,HIGH);
        digitalWrite(9,HIGH);
        digitalWrite(8,HIGH);
    }
}

```