// BRAD //

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//

char stringToMorseCode[] = "Arduino Morse Code Project";

// Create variable to define the output pins

int led12 = 1; // blink an led on output 12

int led6 = 2; // blink an led on output 6

int dotLen = 100; // length of the morse code 'dot'

int dashLen = dotLen \* 3; // length of the morse code 'dash'

int elemPause = dotLen; // length of the pause between elements of a character

int Spaces = dotLen \* 3; // length of the spaces between characters

int wordPause = dotLen \* 7; // length of the pause between words

// the setup routine runs once when you press reset:

void setup() {

// initialize the digital pin as an output for LED lights.

pinMode(led12, OUTPUT);

pinMode(led6, OUTPUT);

}

// Create a loop of the letters/words you want to output in morse code (defined in string at top of code)

void loop()

{

// Loop through the string and get each character one at a time until the end is reached

for (int i = 0; i < sizeof(stringToMorseCode) - 1; i++)

{

// Get the character in the current position

char tmpChar = stringToMorseCode[i];

// Set the case to lower case

tmpChar = toLowerCase(tmpChar);

// Call the subroutine to get the morse code equivalent for this character

GetChar(tmpChar);

}

// At the end of the string long pause before looping and starting again

LightsOff(8000);

}

// DOT

void MorseDot()

{

digitalWrite(led12, HIGH); // turn the LED on

digitalWrite(led6, HIGH);

delay(dotLen); // hold in this position

}

// DASH

void MorseDash()

{

digitalWrite(led12, HIGH); // turn the LED on

digitalWrite(led6, HIGH);

delay(dashLen); // hold in this position

}

// Turn Off

void LightsOff(int delayTime)

{

digitalWrite(led12, LOW); // turn the LED off

digitalWrite(led6, LOW);

delay(delayTime); // hold in this position

}

// \*\*\* Characters to Morse Code Conversion \*\*\* //

void GetChar(char tmpChar)

{

// Take the passed character and use a switch case to find the morse code for that character

switch (tmpChar) {

case 'b':

MorseDash();

LightsOff(elemPause);

MorseDot();

LightsOff(elemPause);

MorseDot();

LightsOff(elemPause);

MorseDot();

LightsOff(elemPause);

break;

case 'r':

MorseDot();

LightsOff(elemPause);

MorseDash();

LightsOff(elemPause);

MorseDot();

LightsOff(elemPause);

break;

case 'a':

MorseDot();

LightsOff(elemPause);

MorseDash();

LightsOff(elemPause);

break;

case 'd':

MorseDash();

LightsOff(elemPause);

MorseDash();

LightsOff(elemPause);

MorseDot();

LightsOff(elemPause);

break;

default:

// If a matching character was not found it will default to a blank space

LightsOff(Spaces);

}

}