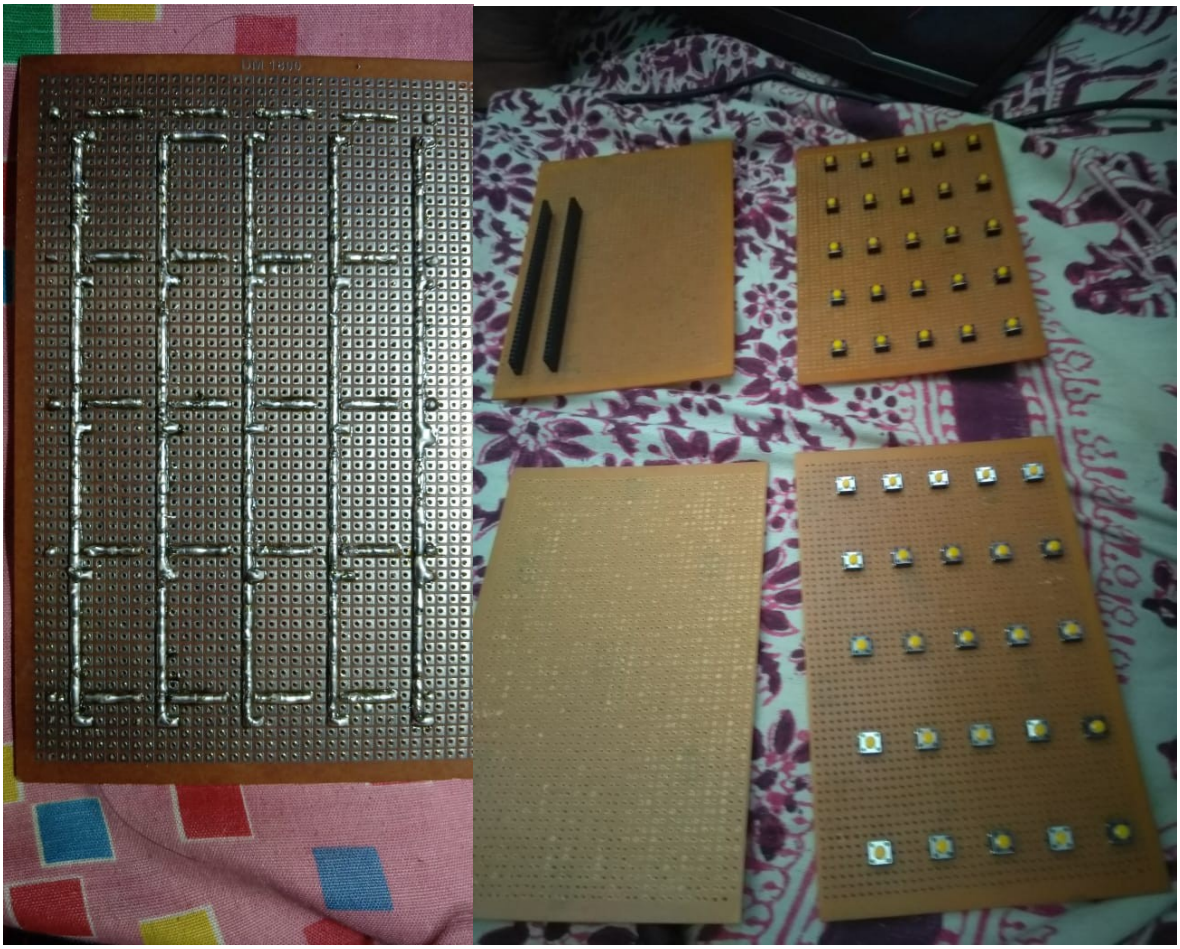
TASKS COMPLETED-

Completed the coding for the keyboard and did the soldering and connections work for the keyboard. Also completed the interphase program coding for the morse code signal transmission. Only physical connections for all the components and dry run task are remaining.

1) ARDUINO CODE FOR KEYBOARD AND CIRCUIT DESIGN-



```
#include <LiquidCrystal.h>


const int buttonPin1 = A0;// the number of the pushbutton pin
const int buttonPin2 = A1;
const int buttonPin3 = A2;
const int buttonPin4 = A3;
```

```
const int buttonPin5 = A4;
const int buttonPin6 = A5;
const int A = 0;
const int B = 1;
const int C = 2;
const int D = 3;
const int E = 4;

int buttonState1;
int buttonState2;
int buttonState3;
int buttonState4;
int buttonState5;
int buttonState6;

const int mic = 5;
const int buzz = 6;
int caps;
int count;

const int rs = 8, en = 9, d4 = 10, d5 = 11, d6 = 12, d7 = 13;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

const int ledPin =  13;      // the number of the LED pin

void setup() {
pinMode(buttonPin1,INPUT);
pinMode(buttonPin2,INPUT);
pinMode(buttonPin3,INPUT);
pinMode(buttonPin4,INPUT);
pinMode(buttonPin5,INPUT);

pinMode(A, OUTPUT);
pinMode(B, OUTPUT);
pinMode(C, OUTPUT);
pinMode(D, OUTPUT);
pinMode(E, OUTPUT);

pinMode(mic,INPUT);
pinMode(buzz,INPUT);



lcd.begin(16, 2);
```

```
lcd.print("MORSE INTERPHASE");

buttonState1 = 0;
buttonState2 = 0;
buttonState3 = 0;
buttonState4 = 0;
buttonState5 = 0;

count=1;
}

void loop() {

// read the state of the pushbutton value:
buttonState1 = digitalRead(buttonPin1);
buttonState2 = digitalRead(buttonPin2);
buttonState3 = digitalRead(buttonPin3);
buttonState4 = digitalRead(buttonPin4);
buttonState5 = digitalRead(buttonPin5);


digitalWrite(A,HIGH);
digitalWrite(B,LOW);
digitalWrite(C,LOW);
digitalWrite(D,LOW);
digitalWrite(E,LOW);


if(buttonState1 == HIGH){
  if(caps){
 // "V" being sent to LCD Display
 lcd.print("V");
  }
 else{
 // "A" being sent to LCD Display
 lcd.print("A");
 }
 count=count+1;
//delay*
 delay(100);
 if(count ==16){lcd.setCursor(0, 1);}  //abcdefg
}

if(buttonState2 == HIGH){
```

```
 if(caps){
// "0" being sent to LCD Display
lcd.print("0");
}
else{
// "F" being sent to LCD Display
lcd.print("F");
}
count=count+1;

 //delay*
delay(100);
if(count ==16){lcd.setCursor(0, 1);}  //abcdefg
}

if(buttonState3 == HIGH){
 if(caps){
// "5" being sent to LCD Display
lcd.print("5");
}
else{
// "K" being sent to LCD Display
lcd.print("K");
}
count=count+1;

//delay*
delay(100);
if(count ==16){lcd.setCursor(0, 1);}  //abcdefg
}

if(buttonState4 == HIGH){
// "P" being sent to LCD Display
lcd.print("P");
count=count+1;

 //delay*
delay(100);
if(count ==16){lcd.setCursor(0, 1);}  //abcdefg
}

if(buttonState5 == HIGH){
// caps is made ON/OFF
 caps=1;
```

```
//delay*
 delay(100);
}

if(buttonState5 == LOW){
 // caps is made ON/OFF
  caps=0;
//delay*
 delay(100);
}

// Second column


digitalWrite(A,LOW);
digitalWrite(B,HIGH);
digitalWrite(C,LOW);
digitalWrite(D,LOW);
digitalWrite(E,LOW);

buttonState1 = digitalRead(buttonPin1);
buttonState2 = digitalRead(buttonPin2);
buttonState3 = digitalRead(buttonPin3);
buttonState4 = digitalRead(buttonPin4);
buttonState5 = digitalRead(buttonPin5);


if(buttonState1 == HIGH){
  if(caps){
 // "W" being sent to LCD Display
 lcd.print("W");
 }
 else{
 // "B" being sent to LCD Display
 lcd.print("B");
 }
 count=count+1;

//delay*
 delay(100);
 if(count ==16){lcd.setCursor(0, 1);}  //abcdefg
}

if(buttonState2 == HIGH){
```

```
 if(caps){
// "1" being sent to LCD Display
lcd.print("1");
 }
else{
// "G" being sent to LCD Display
lcd.print("G");
}
count=count+1;

//delay*
 delay(100);
 if(count ==16){lcd.setCursor(0, 1);}  //abcdefg
}

if(buttonState3 == HIGH){
 if(caps){
// "6" being sent to LCD Display
lcd.print("6");
 }
  // "L" being sent to LCD Display
else{lcd.print("L");}
count=count+1;

//delay*
 delay(100);
 if(count ==16){lcd.setCursor(0, 1);}  //abcdefg
}

if(buttonState4 == HIGH){
 // "Q" being sent to LCD Display
 lcd.print("Q");
 count=count+1;

//delay*

 delay(100);
 if(count ==16){lcd.setCursor(0, 1);}  //abcdefg
}

if(buttonState5 == HIGH){
 // "T" being sent to LCD Display
 lcd.print("T");
 count=count+1;
```

```
//delay*
 delay(100);
 if(count ==16){lcd.setCursor(0, 1);}  //abcdefg
}


//Third Column

digitalWrite(A,LOW);
digitalWrite(B,LOW);
digitalWrite(C,HIGH);
digitalWrite(D,LOW);
digitalWrite(E,LOW);

buttonState1 = digitalRead(buttonPin1);
buttonState2 = digitalRead(buttonPin2);
buttonState3 = digitalRead(buttonPin3);
buttonState4 = digitalRead(buttonPin4);
buttonState5 = digitalRead(buttonPin5);

if(buttonState1 == HIGH){
  if(caps){
 // "X" being sent to LCD Display
 lcd.print("X");

  }
 else{
 // "C" being sent to LCD Display
 lcd.print("C");
 }
 count=count+1;

//delay*
 delay(100);
 if(count ==16){lcd.setCursor(0, 1);}  //abcdefg
}

if(buttonState2 == HIGH){
  if(caps){
 // "2" being sent to LCD Display
 lcd.print("2");
}
 else{
```

```
 // "H" being sent to LCD Display
 lcd.print("H");
 }
 count=count+1;

  //delay*
 delay(100);
 if(count ==16){lcd.setCursor(0, 1);}  //abcdefg
}

if(buttonState3 == HIGH){
  if(caps){
 // "7" being sent to LCD Display
 lcd.print("7");
}
   // "M" being sent to LCD Display
 else {lcd.print("M");}

 count=count+1;
//delay*
 delay(100);
 if(count ==16){lcd.setCursor(0, 1);}  //abcdefg
}

if(buttonState4 == HIGH){
 // "R" being sent to LCD Display
 lcd.print("R");
 count=count+1;

  //delay*
 delay(100);
 if(count ==16){lcd.setCursor(0, 1);}  //abcdefg
}

if(buttonState5 == HIGH){
 // "U" being sent to LCD Display
 lcd.print("U");
 count=count+1;

  //delay*
 delay(100);
 if(count ==16){lcd.setCursor(0, 1);}  //abcdefg
}
```

```
//Fourth Column

digitalWrite(A,LOW);
digitalWrite(B,LOW);
digitalWrite(C,LOW);
digitalWrite(D,HIGH);
digitalWrite(E,LOW);

buttonState1 = digitalRead(buttonPin1);
buttonState2 = digitalRead(buttonPin2);
buttonState3 = digitalRead(buttonPin3);
buttonState4 = digitalRead(buttonPin4);
buttonState5 = digitalRead(buttonPin5);

if(buttonState1 == HIGH){
  if(caps){
 // "Y" being sent to LCD Display
 lcd.print("Y");

  }
 else{
 // "D" being sent to LCD Display
 lcd.print("D");
 }

 count=count+1;
//delay*
 delay(100);
 if(count ==16){lcd.setCursor(0, 1);}  //abcdefg
}

if(buttonState2 == HIGH){
  if(caps){
 // "3" being sent to LCD Display
 lcd.print("3");
  }
 else{
 // "I" being sent to LCD Display
 lcd.print("I");
 }

 count=count+1;
 //delay*
```

```
 delay(100);
 if(count ==16){lcd.setCursor(0, 1);}  //abcdefg
}

if(buttonState3 == HIGH){
   if(caps){
 // "8" being sent to LCD Display
 lcd.print("8");
  }
 else{
 // "N" being sent to LCD Display
 lcd.print("N");
 }
//delay*
 delay(100);
 if(count ==16){lcd.setCursor(0, 1);}  //abcdefg
}

if(buttonState4 == HIGH){
 // "S" being sent to LCD Display
 lcd.print("S");
 count=count+1;

 //delay*

 delay(100);
 if(count ==16){lcd.setCursor(0, 1);}  //abcdefg
}

if(buttonState5 == HIGH){
 // " " being sent to LCD Display
 lcd.print(" ");

 count=count+1;
 //delay*
 delay(100);
 if(count ==16){lcd.setCursor(0, 1);}  //abcdefg
}


// Fifth Column

digitalWrite(A,LOW);
digitalWrite(B,LOW);
```

```
digitalWrite(C,LOW);
digitalWrite(D,LOW);
digitalWrite(E,HIGH);

buttonState1 = digitalRead(buttonPin1);
buttonState2 = digitalRead(buttonPin2);
buttonState3 = digitalRead(buttonPin3);
buttonState4 = digitalRead(buttonPin4);
buttonState5 = digitalRead(buttonPin5);

if(buttonState1 == HIGH){
  if(caps){
 // "Z" being sent to LCD Display
 lcd.print("Z");
  }
 else{
 // "E" being sent to LCD Display
 lcd.print("E");
 }

 count=count+1;
//delay*
 delay(100);
 if(count ==16){lcd.setCursor(0, 1);}  //abcdefg
}

if(buttonState2 == HIGH){
  if(caps){
 // "4" being sent to LCD Display
 lcd.print("4");
 }
 else{
 // "J" being sent to LCD Display
 lcd.print("J");
 }

 count=count+1;
 //delay*
 delay(100);
 if(count ==16){lcd.setCursor(0, 1);}  //abcdefg
}

if(buttonState3 == HIGH){
   if(caps){
```

```
 // "9" being sent to LCD Display
 lcd.print("9");
 }
 else{
 // "O" being sent to LCD Display
 lcd.print("O");
 }

 count=count+1;
//delay*
 delay(100);
 if(count ==16){lcd.setCursor(0, 1);}  //abcdefg
}


// Edit this one for BACKSPACE
if(buttonState4 == HIGH){
 // "T" being sent to LCD Display
 lcd.print("T");

 count=count+1;
//delay*

 delay(100);
 if(count ==16){lcd.setCursor(0, 1);}  //abcdefg
}



// Edit this one for ENTER

if(buttonState5 == HIGH){
 // "X" being sent to LCD Display
 lcd.print("X");
//delay*
 delay(100);
 if(count ==16){lcd.setCursor(0, 1);}  //abcdefg
}


}
```

2) CODE FOR CONVERTING MESSAGE TO MORSE CODE-

```
/* Serial Morse Encoder and Decoder
 *  written by 'jurs' for Arduino forum
 *  Usage:
 *  Use the serial monitor to send ASCII text ==> will be encoded to morse code
 *  Use the serial monitor to send dots and dashes ==> will be decoded to ASCII
 */

char morseCode[]={ // first bit set tells where morse encoding starts
  B00000101, // A = .-
  B00011000, // B = -...
  B00011010, // C = -.-.
  B00001100, // D = -..
  B00000010, // E = .
  B00010010, // F = ..-.
  B00001110, // G = --.
  B00010000, // H = ....
  B00000100, // I = ..
  B00010111, // J = .---
  B00001101, // K = -.-
  B00010100, // L = .-..
  B00000111, // M = --
  B00000110, // N = -.
  B00001111, // O = ---
  B00010110, // P = .--.
  B00011101, // Q = --.-
  B00001010, // R = .-.
  B00001000, // S = ...
  B00000011, // T = -
  B00001001, // U = ..-
  B00010001, // V = ...-
  B00001011, // W = .--
  B00011001, // X = -..-
  B00011011, // Y = -.--
  B00011100, // Z = --..
  B00111111, // 0 = -----
  B00101111, // 1 = .----
  B00100111, // 2 = ..---
  B00100011, // 3 = ...--
  B00100001, // 4 = ....-
  B00100000, // 5 = .....
  B00110000, // 6 = -....
  B00111000, // 7 = --...
  B00111100, // 8 = ---..
  B00111110, // 9 = ----.
};
```

```
#define NUMCHARS sizeof(morseCode)

char morseChars[]="ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789";

void sendMorseChar(char c)
{
  // invalid ASCII chars will not be encoded into anything
  // valid ASCII chars are encoded into morse dots/dashes
  // then finish with sending a space character ' '
  char* found= strchr(morseChars, c);
  if (found==NULL) return;
  else
  {
    byte morse= morseCode[found-morseChars];
    boolean firstBitFound=false;
    for (int i=7; i>= 0; i--)
    {
      byte thisBit= bitRead(morse,i);
      if (firstBitFound)
      {
        if (thisBit) Serial.print('-');
        else Serial.print('.');
      }
      else if (thisBit) firstBitFound=true;
    }
  }
  Serial.print(" ");
}

char handleMorseInput(char c)
{ // char c==0 is for clearing internal receiveBuf
  // char c==1 is for evaluating internal receiveBuf
  static byte receiveBuf;
  if (c==0)
  {
    receiveBuf=1; // set 0-bit
    return 0;
  }
  else if (c==1)
  {
    for (int i=0;i<NUMCHARS;i++)
    {
      if (receiveBuf==morseCode[i])
      {
        return morseChars[i];
```

```
      }
    }
    return('*');
  }
  else if (c=='.') receiveBuf= receiveBuf<<1;
  else if (c=='-') receiveBuf= (receiveBuf<<1) | 1;
  return 0;
}

void handleSerialInput()
{
  static boolean isMorseInput=false;
  if (!Serial.available()) return;
  char c=Serial.read();
  // automatic detection of morse or ASCII input
  if (c=='.' || c=='-')
  {
    if (!isMorseInput)
    { // switch to morse input and clear input buffer
      isMorseInput= true;
      handleMorseInput(0);
    }
    handleMorseInput(c);
  }
  else
  {
    if (isMorseInput)
    {
      c= handleMorseInput(1); // retrieve decoded char from input buffer
      Serial.print(c);
      isMorseInput=false;
    }
    else if (c==' ') Serial.print(' ');
    else if (c=='\n') Serial.println();
    else if (c>32) // filter out control characters and international special characters
    {
      sendMorseChar(toupper(c));
    }
  }
}

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

void loop() {
```

```
  handleSerialInput();
}
```