```
//Modified from examples by Pat McMahon 8/11/2020
// Use the MD_MAX72XX library to scroll text on the display
//
//
// Demonstrates the use of the callback function to control what
// is scrolled on the display text.
//
// User can enter text on the serial monitor and this will display as a
// scrolling message on the display.
// Speed for the display is controlled by a pot on SPEED_IN analog in.
//
#include <MD_MAX72xx.h>
#include <SPI.h>

#define USE_POT_CONTROL 1
#define PRINT_CALLBACK  0

#define PRINT(s, v) { Serial.print(F(s)); Serial.print(v); }

// Define the number of devices we have in the chain and the hardware
interface
// NOTE: These pin numbers will probably not work with your hardware and may
// need to be adapted
#define HARDWARE_TYPE MD_MAX72XX::DR1CR0RR0_HW
#define MAX_DEVICES 20


#define CLK_PIN   13  // or CLK
#define DATA_PIN  11  // or DIN
#define CS_PIN    10  // or CS

// SPI hardware interface
MD_MAX72XX mx = MD_MAX72XX(HARDWARE_TYPE, CS_PIN, MAX_DEVICES);
// Arbitrary pins
//MD_MAX72XX mx = MD_MAX72XX(HARDWARE_TYPE, DATA_PIN, CLK_PIN, CS_PIN,
MAX_DEVICES);

// Scrolling parameters
#if USE_POT_CONTROL
#define SPEED_IN  A0
#else
#define SCROLL_DELAY  75  // in milliseconds
#endif // USE_POT_CONTROL

#define CHAR_SPACING  1 // pixels between characters

// Global message buffers shared by Serial and Scrolling functions
#define BUF_SIZE 300
```

```
char curMessage[BUF_SIZE];
char newMessage[BUF_SIZE];
bool newMessageAvailable = false;

uint16_t  scrollDelay;  // in milliseconds

void readSerial(void)
{
  static uint8_t  putIndex = 0;

  while (Serial.available())
  {
    newMessage[putIndex] = (char)Serial.read();
    if ((newMessage[putIndex] == '\n') || (putIndex >= BUF_SIZE-3)) // end of
message character or full buffer
    {
      // put in a message separator and end the string
      newMessage[putIndex++] = ' ';
      newMessage[putIndex] = '\0';
      // restart the index for next filling spree and flag we have a message
waiting
      putIndex = 0;
      newMessageAvailable = true;
    }
    else if (newMessage[putIndex] != '\r')
      // Just save the next char in next location
      putIndex++;
  }
}

void scrollDataSink(uint8_t dev, MD_MAX72XX::transformType_t t, uint8_t col)
// Callback function for data that is being scrolled off the display
{
#if PRINT_CALLBACK
  Serial.print("\n cb ");
  Serial.print(dev);
  Serial.print(' ');
  Serial.print(t);
  Serial.print(' ');
  Serial.println(col);
#endif
}

uint8_t scrollDataSource(uint8_t dev, MD_MAX72XX::transformType_t t)
// Callback function for data that is required for scrolling into the display
{
  static char   *p = curMessage;
  static uint8_t  state = 0;
```

```
  static uint8_t  curLen, showLen;
  static uint8_t  cBuf[8];
  uint8_t colData;

  // finite state machine to control what we do on the callback
  switch(state)
  {
    case 0: // Load the next character from the font table
      showLen = mx.getChar(*p++, sizeof(cBuf)/sizeof(cBuf[0]), cBuf);
      curLen = 0;
      state++;

      // if we reached end of message, reset the message pointer
      if (*p == '\0')
      {
        p = curMessage;     // reset the pointer to start of message
        if (newMessageAvailable)  // there is a new message waiting
        {
          strcpy(curMessage, newMessage); // copy it in
          newMessageAvailable = false;
        }
      }
      // !! deliberately fall through to next state to start displaying

    case 1: // display the next part of the character
      colData = cBuf[curLen++];
      if (curLen == showLen)
      {
        showLen = CHAR_SPACING;
        curLen = 0;
        state = 2;
      }
      break;

    case 2: // display inter-character spacing (blank column)
      colData = 0;
      curLen++;
      if (curLen == showLen)
        state = 0;
      break;

    default:
      state = 0;
  }

  return(colData);
}
```

```
 void scrollText(void)
{
  static uint32_t prevTime = 0;

  // Is it time to scroll the text?
  if (millis()-prevTime >= scrollDelay)
  {
    mx.transform(MD_MAX72XX::TSL);  // scroll along - the callback will load
all the data
    prevTime = millis();      // starting point for next time
  }
}

uint16_t getScrollDelay(void)
{
#if USE_POT_CONTROL
  uint16_t  t;

  t = analogRead(SPEED_IN);
  t = map(t, 0, 1023, 25, 250);

  return(t);
#else
  return(SCROLL_DELAY);
#endif
}

void setup()
{
  mx.begin();
  mx.setShiftDataInCallback(scrollDataSource);
  mx.setShiftDataOutCallback(scrollDataSink);

#if USE_POT_CONTROL
  pinMode(SPEED_IN, INPUT);
#else
  scrollDelay = SCROLL_DELAY;
#endif

strcpy(curMessage, "Welcome to the Whittlesea Tech School, Final Working Day
for the Term 4 Student Ambassador Program 2023. The Student Ambassador
Exhibition Day will be on Thursday the 7th of December 2023, details to
come.              ");


  Serial.begin(57600);
  Serial.print("\n[MD_MAX72XX Message Display]\nType a message for the
scrolling display\nEnd message line with a newline");
}
```

```
void loop()
{
  scrollDelay = getScrollDelay();
  readSerial();
  scrollText();
}
```