```
#include "helper_functions.h"

#ifndef MATRICES_PER_DISPLAY
#include "ff_scoreboard_properties.h"
#endif


// a string to hold incoming data
// the following must be global due to the usage of the serialEvent() function
String inputString = "";
char outputString[100];
boolean stringComplete = false; // whether the string is complete

extern const int pixelPin[];
extern const int matrixPin[];

 // set time by reading serial
extern int glyphIndex[];
// determines location of time digits within input string
// maps 1:1 with glyphIndex
extern const int numberIndexInString[];
// binary mappings of lit and unlit pixels for display.
// 22 bits map to 28 pixels
extern const uint32_t glyphs[];

void setup() {
  // put your setup code here, to run once:

  Serial.begin(115200);
  if (DEBUG) { Serial.println("Serial connection initialized."); }

  // Set all pins in use to low at start, to start with a blank slate
  setMuxingPinsLow();

}

void loop() {
  // put your main code here, to run repeatedly:

  if (stringComplete) {

    if (DEBUG) {
      if (VERBOSE) { Serial.println("Reached top of loop"); }
      Serial.println(inputString);
    }

    stringComplete = (inputStringValid(inputString) && timeStringValid(inputString));
    // now string resembles '+[0-:][0-:]:[0-:][0-:]\n'
    if (stringComplete == true) {
      setGlyphIndex(&glyphIndex[0], inputString, 1);
      setGlyphIndex(&glyphIndex[1], inputString, 2);
      setGlyphIndex(&glyphIndex[2], inputString, 4);
      setGlyphIndex(&glyphIndex[3], inputString, 5);
    }

    if (DEBUG) {
      sprintf(outputString, "Setting matrices to show %d%d:%d%d given ", glyphIndex[0],
glyphIndex[1], glyphIndex[2], glyphIndex[3]);
      Serial.println((outputString+inputString));
      Serial.println();
    }
    // clear the string:
    inputString = "";
    stringComplete = false;
  }

  drawDigitalDisplay(glyphIndex);
}
```

```
// serial polled only if this interrupt handler is written
// serial polled using this handler each time after loop() runs
// needs to be in this main file to run
void serialEvent() {
  while (Serial.available()) {
    // get the new byte:
    char inChar = (char)Serial.read();
    // add it to the inputString:
    inputString += inChar;
    // if the incoming character is a newline, set a flag
    // so the main loop can do something about it:
    if (inChar == '\n') {
      stringComplete = true;
    }
  }
}
```