Supplementary Material

S1 –Programming code used to control sensors: (A) AS7341, (B) AS7262, and (C) TCS34725 and the connection diagrams of the sensors: (D) AS7341, (E) AS7262, and (F) TCS34725.

(A)

```
#include <Adafruit_AS7341.h>
Adafruit_AS7341 as7341;
void setup() {
Serial.begin(115200);
while (!Serial) {
delay(1);
}
if (!as7341.begin()){
Serial.println("Could not find AS7341");
while (1) { delay(10); }
}
as7341.setATIME(100);
as7341.setASTEP(999);
as7341.setGain(AS7341_GAIN_16X); //Adjust the gain to the allowable values (refer to the specifications).
}
void loop() {
if (!as7341.readAllChannels()){
Serial.println("Error reading all channels!");
return;
}
Serial.print("F1 415nm : ");
Serial.println(as7341.getChannel(AS7341_CHANNEL_415nm_F1));
Serial.print("F2 445nm : ");
Serial.println(as7341.getChannel(AS7341_CHANNEL_445nm_F2));
Serial.print("F3 480nm : ");
Serial.println(as7341.getChannel(AS7341_CHANNEL_480nm_F3));
Serial.print("F4 515nm : ");
Serial.println(as7341.getChannel(AS7341_CHANNEL_515nm_F4));
Serial.print("F5 555nm : ");
Serial.println(as7341.getChannel(AS7341_CHANNEL_555nm_F5));
Serial.print("F6 590nm : ");
Serial.println(as7341.getChannel(AS7341_CHANNEL_590nm_F6));
Serial.print("F7 630nm : ");
Serial.println(as7341.getChannel(AS7341_CHANNEL_630nm_F7));
Serial.print("F8 680nm : ");
Serial.println(as7341.getChannel(AS7341_CHANNEL_680nm_F8));
```

(B)

```
#include <Wire.h>
#include "Adafruit_AS726x.h"
// Create the object
Adafruit_AS726x ams;
const int buttonPin = 2;
int buttonState = 0;
// Buffer to hold raw values
uint16_t sensorValues[AS726x_NUM_CHANNELS];
// Buffer to hold calibrated values (not used by default in this example)
// float calibratedValues[AS726x_NUM_CHANNELS];
void setup() {
  Serial.begin(9600);
  pinMode(buttonPin, INPUT);
  while (!Serial);
    // Initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
  // Begin and make sure we can talk to the sensor
  if (!ams.begin()) {
    Serial.println("Could not connect to sensor! Please check your wiring.");
    while (1);
  }
}
void loop() {
  // Read the device temperature
  uint8_t temp = ams.readTemperature();
    // ams.drvOff(); // Uncomment this if you want to use the driver LED for readings
  ams.drvOn();
  ams.setDrvCurrent(0); // 0 = 12.5 mA; 1 = 25 mA; 2 = 50 mA; 3 = 100 mA
  ams.setGain(0); // 0 = 1x; 1 = 3.7x; 2 = 16x; 3 = 64x
  ams.startMeasurement(); // Begin a measurement
    // Wait till data is available
  bool rdy = false;
  while (!rdy) {
    delay(5);
    rdy = ams.dataReady();
  }

  // ams.drvOff(); // Uncomment this if you want to use the driver LED for readings
  // buttonState = digitalRead(buttonPin);
  // if (buttonState == HIGH) { // Comment to go back to continuous method
```

```
Serial.print("Clear    : ");
Serial.println(as7341.getChannel(AS7341_CHANNEL_CLEAR));
Serial.print("Near IR  : ");
Serial.println(as7341.getChannel(AS7341_CHANNEL_NIR));
Serial.println("");
Serial.println("0 mA LED blink");
as7341.setLEDCurrent(100); // The number in parentheses determines the LED current, in this case, it is 100, the maximum allowed.
as7341.enableLED(true);
}
```

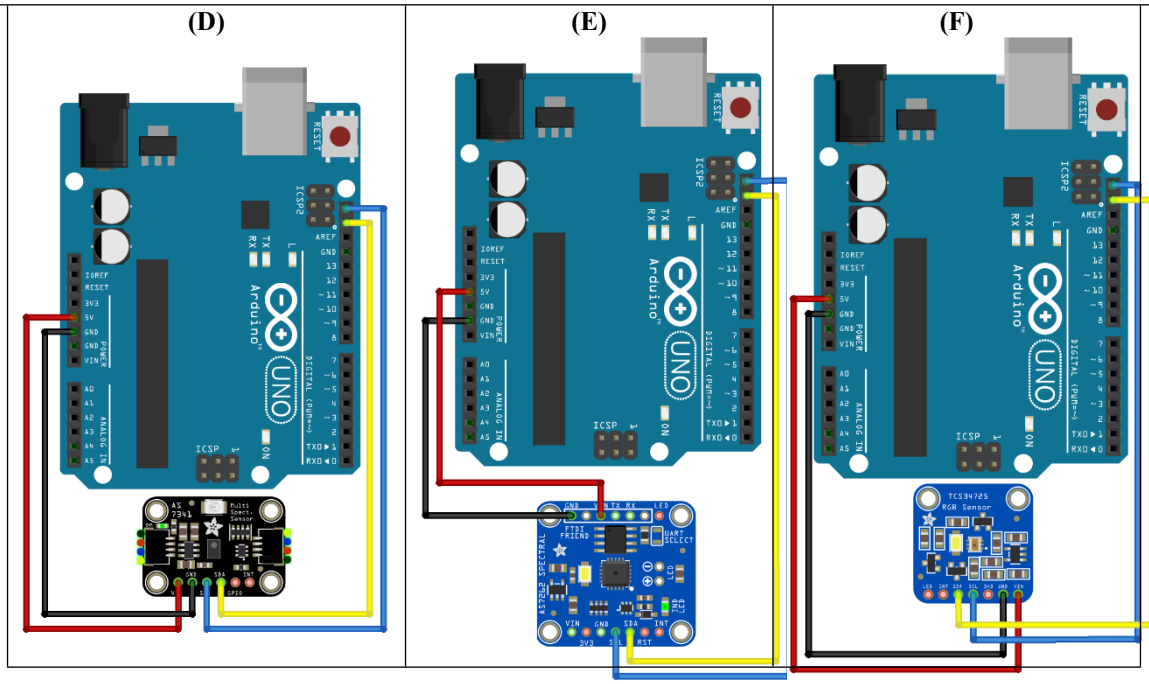
```
  // Read the values!

  ams.readRawValues(sensorValues);
  // ams.readCalibratedValues(calibratedValues);
  Serial.print("Temp: "); Serial.print(temp);
  Serial.print(" Violet: "); Serial.print(sensorValues[AS726x_VIOLET]);
  Serial.print(" Blue: "); Serial.print(sensorValues[AS726x_BLUE]);
  Serial.print(" Green: "); Serial.print(sensorValues[AS726x_GREEN]);
  Serial.print(" Yellow: ");
Serial.print(sensorValues[AS726x_YELLOW]);
  Serial.print(" Orange: ");
Serial.print(sensorValues[AS726x_ORANGE]);
  Serial.print(" Red: "); Serial.print(sensorValues[AS726x_RED]);
  Serial.println();
  Serial.println();
  // } // Comment this to go back to continuous method
}
```

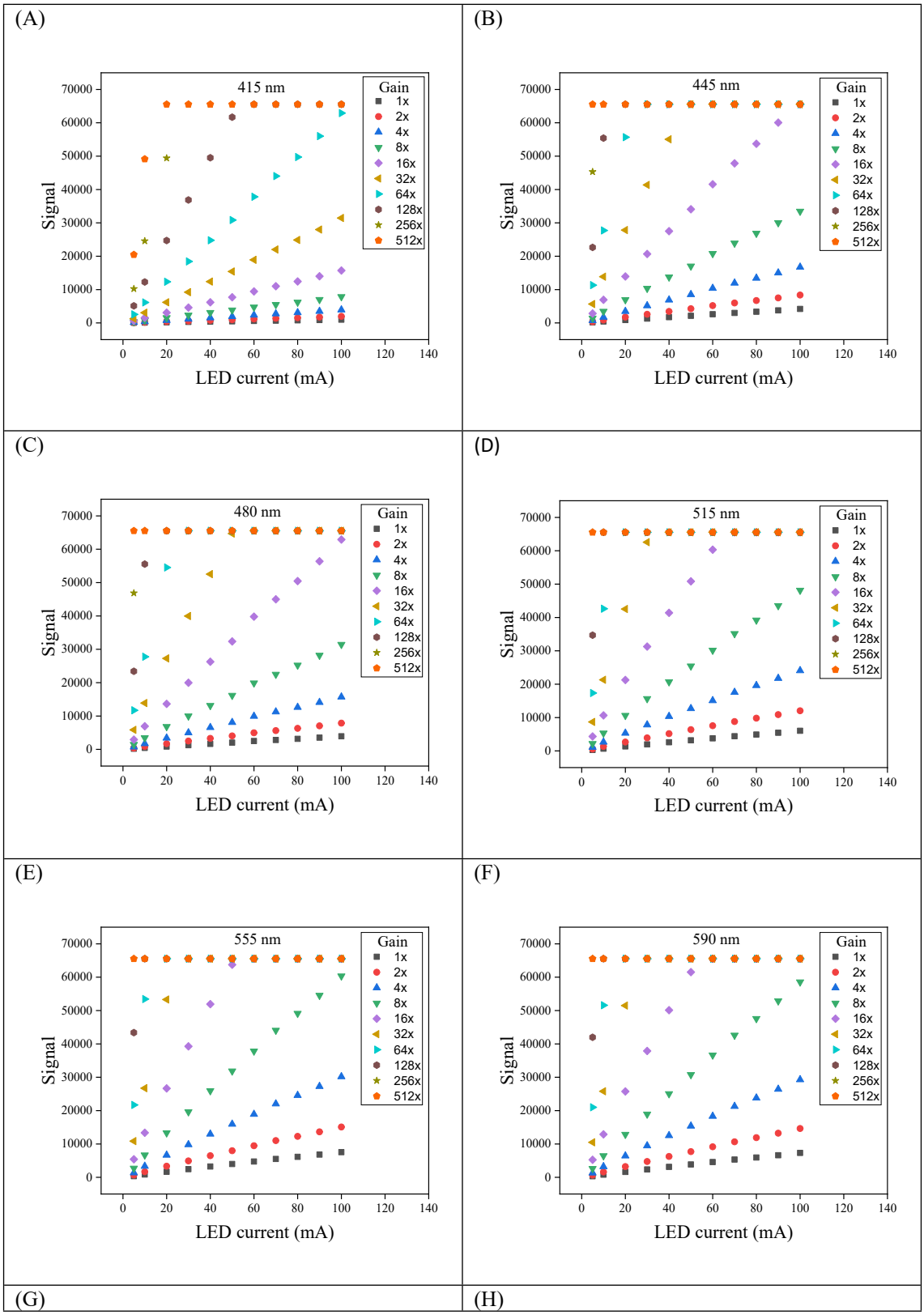
(C)

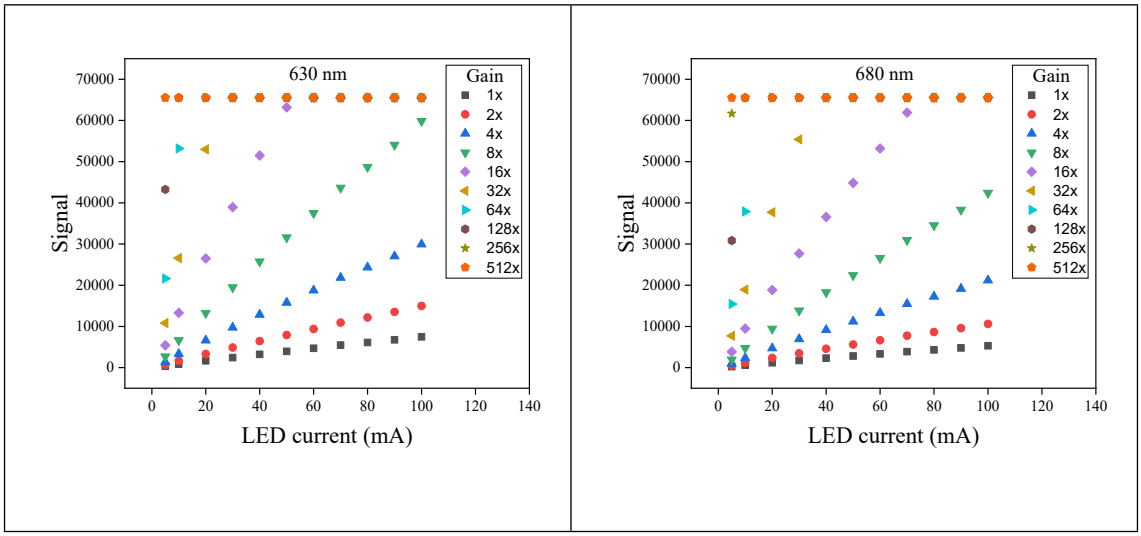
```
/* Initialise with default values (int time = 2.4ms, gain = 1x) */
// Adafruit_TCS34725 tcs = Adafruit_TCS34725();
/* Initialise with specific int time and gain values */
Adafruit_TCS34725 tcs = Adafruit_TCS34725(TCS34725_INTEGRATIONTIME_700MS, TCS34725_GAIN_1X);
void setup(void) {
  Serial.begin(9600);
  if (tcs.begin()) {
    Serial.println("Found sensor");
  } else {
    Serial.println("No TCS34725 found ... check your connections");
    while (1);
  }
  // Now we're ready to get readings!
}
void loop(void) {
  uint16_t r, g, b, c, colorTemp, lux;
  tcs.getRawData(&r, &g, &b, &c);
  // colorTemp = tcs.calculateColorTemperature(r, g, b);
  colorTemp = tcs.calculateColorTemperature_dn40(r, g, b, c);
  lux = tcs.calculateLux(r, g, b);
  Serial.print("Color Temp: ");
  Serial.print(colorTemp, DEC);
  Serial.print(" K - ");
  Serial.print("Lux: ");
  Serial.print(lux, DEC);
  Serial.print(" - ");
  Serial.print("R: ");
  Serial.print(r, DEC);
  Serial.print(" ");
  Serial.print("G: ");
  Serial.print(g, DEC);
  Serial.print(" ");
  Serial.print("B: ");
  Serial.print(b, DEC);
```

```
  Serial.print(" ");
  Serial.print("C: ");
  Serial.print(c, DEC);
  Serial.print(" ");
  Serial.println(" ");
}
```

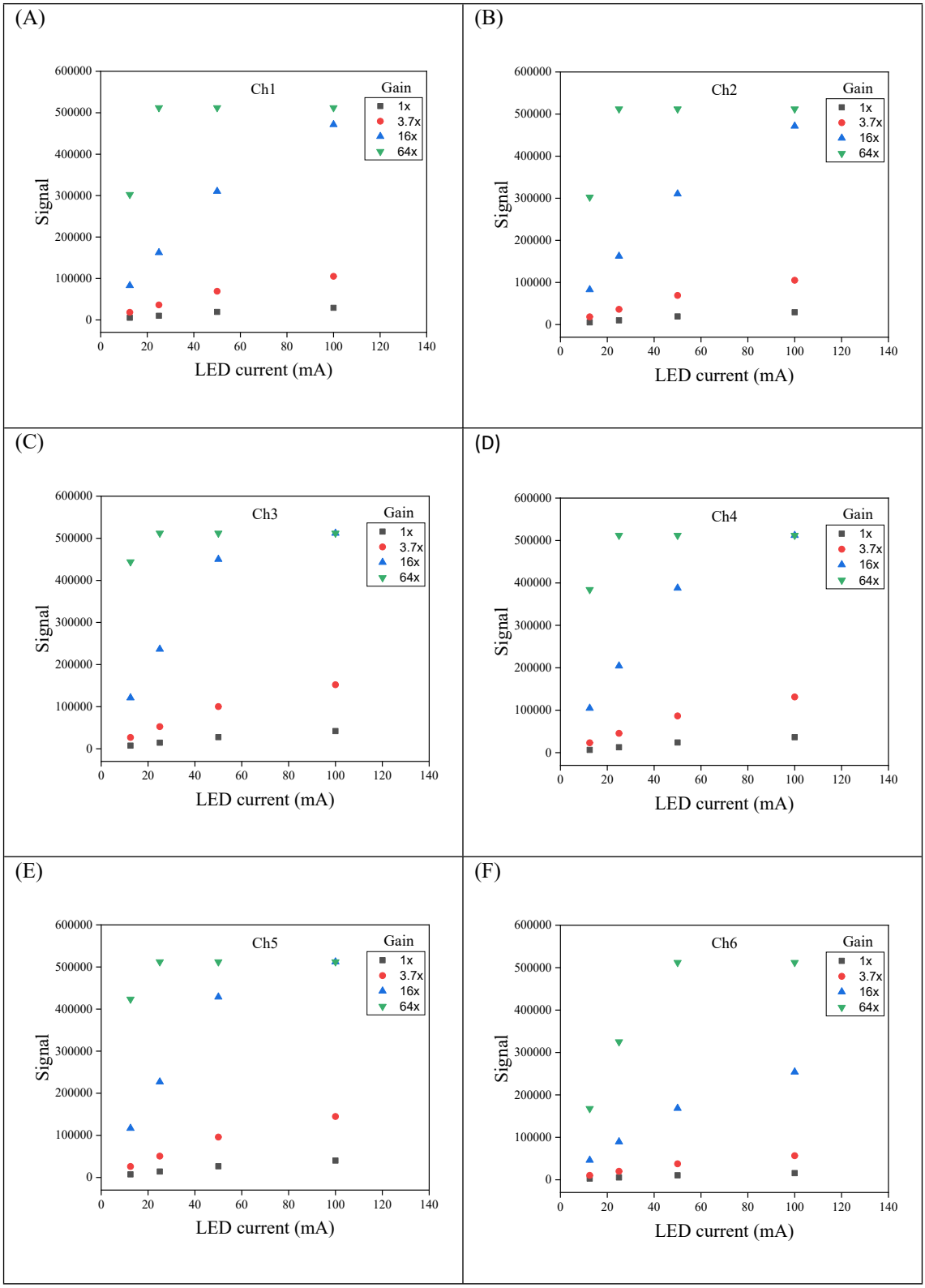


S2 – Evaluation of the effect of gain and LED current for the AS7341 sensor for channels (A) channel 1, (B) channel 2, (C) channel 3, (D) channel 4, (E) channel 5, (F) channel 6, (G) channel 7, and (H) channel 8.




630 nm
Signal
LED current (mA)
Gain
1x
2x
4x
8x
16x
32x
64x
128x
256x
512x
680 nm
Signal
LED current (mA)
Gain
1x
2x
4x
8x
16x
32x
64x
128x
256x
512x

S3 - Evaluation of the effect of gain and LED current for the AS7262 sensor for channels (A) channel 1, (B) channel 2, (C) channel 3, (D) channel 4, (E) channel 5, (F) channel 6.



S4 – Calibration curves for sunset yellow (■) and sunset yellow in the presence of equimolar tartrazine (●).