

Review of Hamamastu C12880 Microspec Module

Craig Versek (Northeastern University)
Parker Woodworth (Food+Future)
Alex Meckes (Wyss Institute)
Don Blair (Food+Future)

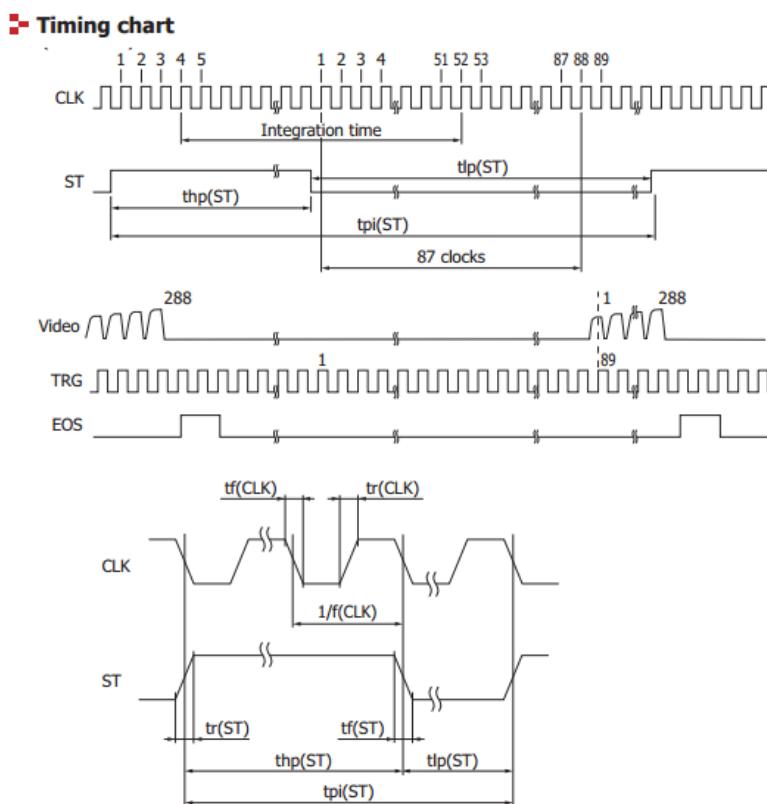
Summary

The C12880MA has is an easy-to-use, highly portable sensor module, useful in the range of x to y nm. We believe that the currently available datasheet for the C12880MA gives the impression that the device requires more sophisticated hardware than is in fact needed in order to effectively operate the device in a wide range of applications; two independent engineering teams initially avoided using it because it seems to require a protocol speed only available on custom FPGA chips. Below, we demonstrate a basic application of the C12880MA that uses a \$8 hobby microcontroller to effectively produce transmission spectra. We recommend that the language in the datasheet and advertising be modified to reflect how truly accessible the module is, thus making the module immediate more appealing to a wider audience.

First impression: perhaps we're too slow?

In our evaluation of the C12880MA module, we relied on the datasheet available [here](#).

The timing chart on page 7 of the C12880MA datasheet strongly suggests that a regular clock pulse is needed at all times in order to operate the device.



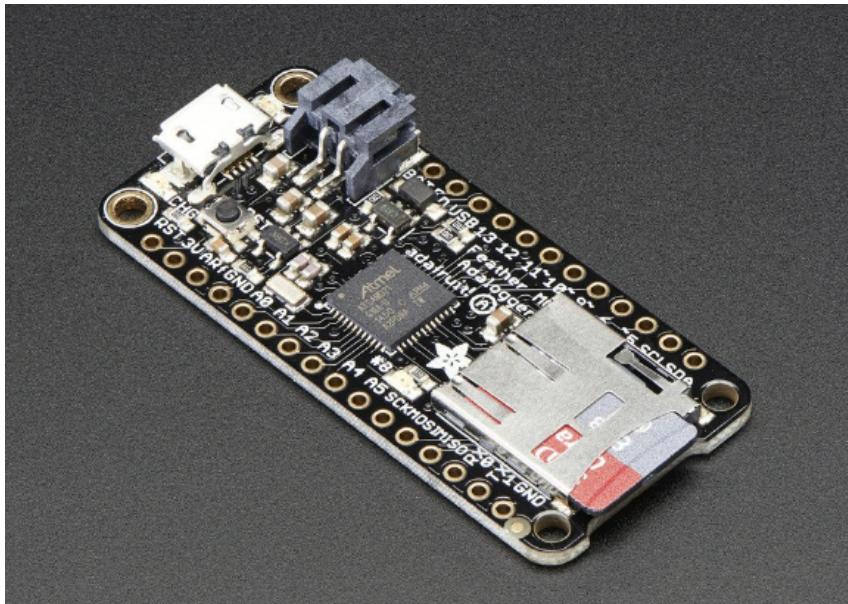
Our first impression (shared with another engineering team at MIT) was that reading off pixel values from the C12880MA with a microcontroller ADC would therefore need to be performed within this ongoing clock pulse schedule -- a constraint that would have required a faster and more advanced microcontroller architecture than any of us was comfortable developing.

We eventually discovered [example code](#), provided by [GroupGets](#), for operating the C12880MA using the Arduino microcontroller. After successfully running this code, we realized that we had misinterpreted the datasheet: after an initial sequence of pulses to prepare the device and set the integration time, the C12880MA then allows individual pixel values to be read off via the ADC at whatever pace is convenient -- a constant clock pulse is not required.

After discovering and studying this example, we were able to write our own C and Python code (described below) to control the device.

Low-cost, accessible microcontroller setup for the C12880MA

Here we present code and hardware for interfacing with the C12880MA using low-cost microcontrollers. The code we developed is compatible with a wide range of microcontrollers that utilize the [Arduino platform](#). Our setup uses the [Adafruit Feather](#):



-- but our code is also compatible with:

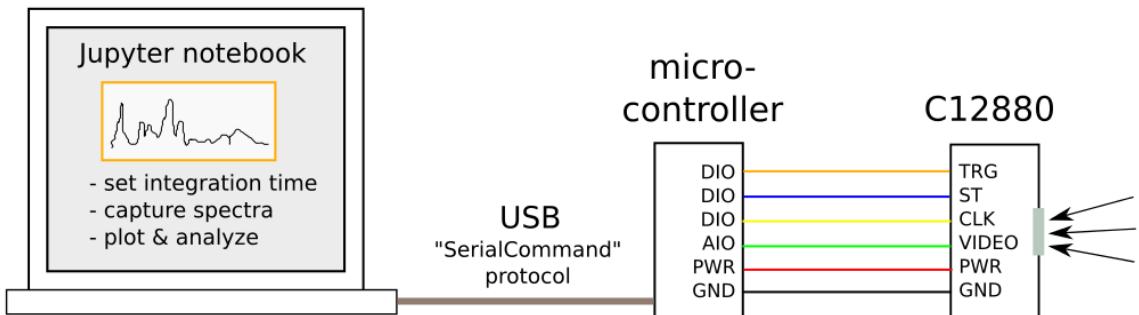
- [Adafruit Feather](#) (\$15) (tested)
- [Arduino Pro Mini](#) (\$10) (tested)
- The [Teensy 3.2](#) (\$20) (tested)
- Other microcontrollers compatible with the [Arduino IDE](#) (untested)

Wiring diagram

Our basic setup involves using a low-cost microcontroller to interface directly with the C12880MA. Such a microcontroller can then be programmed to store spectral data locally to an SD card, broadcast spectral data over radio, or send values over a serial connection.

We opted for the latter option: we used the [SerialCommand](#) protocol to send commands to the microcontroller, via a Python script that interacts with the relevant serial port on a laptop. The laptop is connected via USB serial to the microcontroller; the microcontroller is connected via wires to the microspec module, as per the C12880MA datasheet:

laptop / PC / Raspberry Pi



Serial control via Python; Jupyter notebook analysis

Our [Custom firmware](#) consists of Arduino code to control the C12880MA via its custom protocol:

- [c12880.cpp](#)
- [c12880.h](#)

As well as Arduino code to wrap this protocol, and allow for serial communication over USB:

- [microscpec.ino](#)

And, finally, Python code to interact with the serial protocol:

- [test.py](#)

First Test: florescent bulb

Our first test of the C12880 was using a fluorescent bulb, to see if we could capture a characteristic spectrum. We used a [Jupyter notebook for capturing and analyzing the spectrum](#). The notebook allows for running the Python code above, which interacts with the C12880 module via USB.

First, the notebook imports the relevant modules, and sets the integration time on the C12880 device:

```
In [2]: import os, glob, serial
from pylab import *

In [3]: class MicroSpec(object):
    def __init__(self, port):
        self._ser = serial.Serial(port, baudrate=115200)
    def set_integration_time(self, seconds):
        cmd = "SPEC.INTEG %0.6f\n" % seconds
        self._ser.write(cmd.encode('utf8'))
    def read(self):
        self._ser.write(b"SPEC.READ?\n")
        sdata = self._ser.readline()
        sdata = array([int(p) for p in sdata.split(b",")])
        self._ser.write(b"SPEC.TIMING?\n")
        tdata = self._ser.readline()
        tdata = array([int(p) for p in tdata.split(b",")])
        return (sdata, tdata)

DATASET_NAME = "Xenon-20-watt_no-filt_3inch-water_integ50us"
#mpl.rcParams["savefig.directory"] = os.path.dirname(__file__)

port = glob.glob("/dev/ttyACM*")[0]

spec = MicroSpec(port)
icc = spec.set_integration_time(100e-3)
```

Then, we generating the proper wavelength vector according to the calibration fitting parameters provided by Hamamastu, and capture and plot the spectrum:

```
In [50]: pix=arange(1,289);

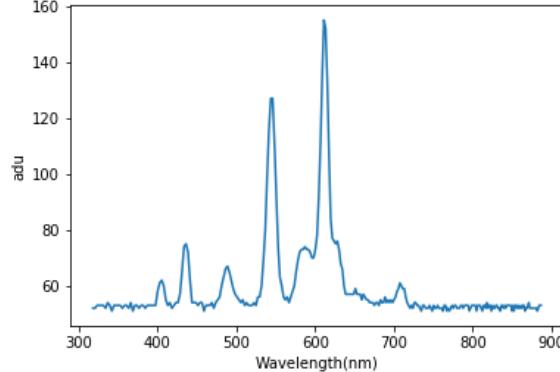
A_0=3.152446842e+2; B_1=2.688494791; B_2=-8.964262020e-4;
B_3=-1.030880174e-5; B_4=2.083514791e-8; B_5=-1.290505933e-11;

nm=A_0+B_1*pix+B_2*pix**2+B_3*pix**3+B_4*pix**4+B_5*pix**5;

In [48]: sdata, tdata = spec.read()

In [49]: plot(nm,sdata);
xlabel('Wavelength(nm)')
ylabel('adu')
```

Out[49]: <matplotlib.text.Text at 0x7f0ee48b03c8>

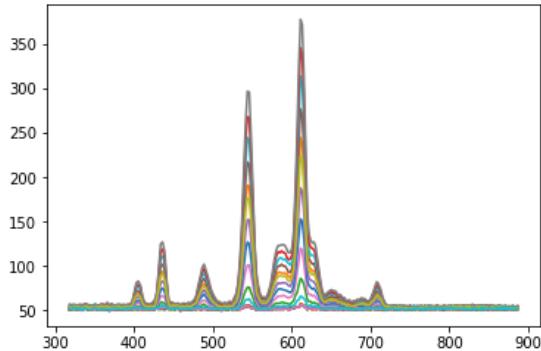


Varying the integration time

We also used a similar setup in [another Jupyter notebook](#) to test our ability to vary the integration time in a straightforward way:

```
In [121]: y,times=integration_plot(.1,1.1,.05)
```

```
In [122]: for i in range(0,len(y)):
    plot(nm,y[i]);
```

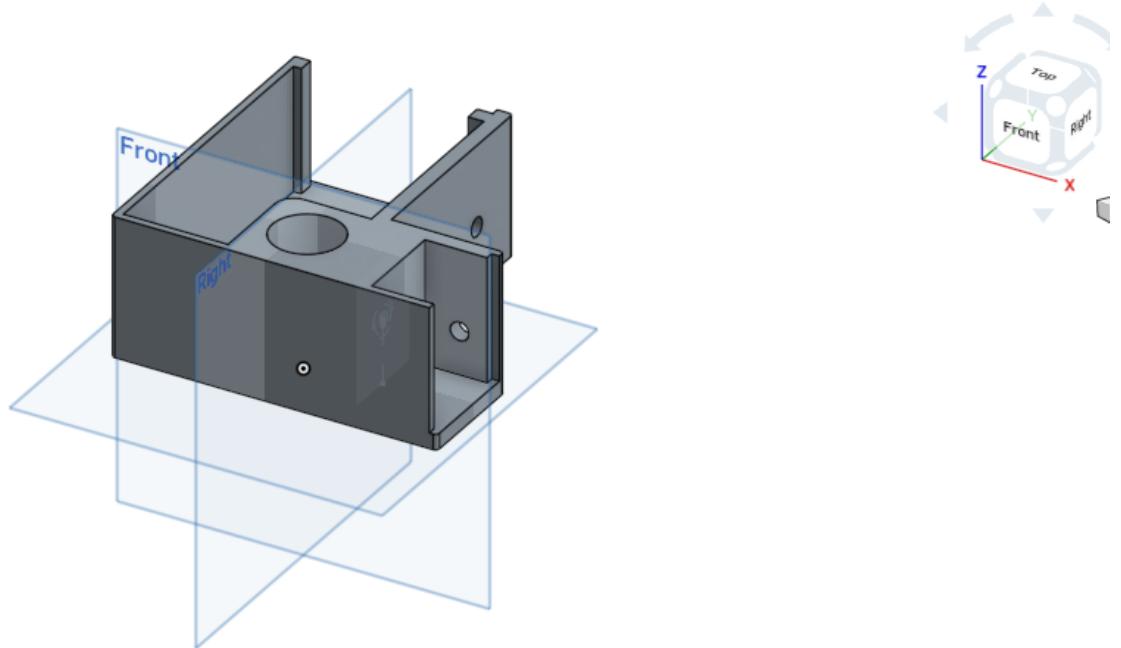


3D Printed Transmission Spectroscopy Enclosure for the C12880MA

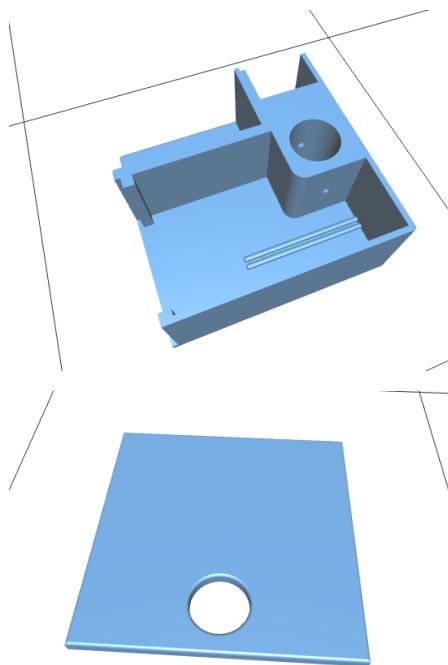
In order to perform some basic transmission spectroscopy experiments, we designed and printed a 3D enclosure for a protoboard version of our C12880 setup. The enclosure accommodates a vial that can be found on Amazon [here](#).



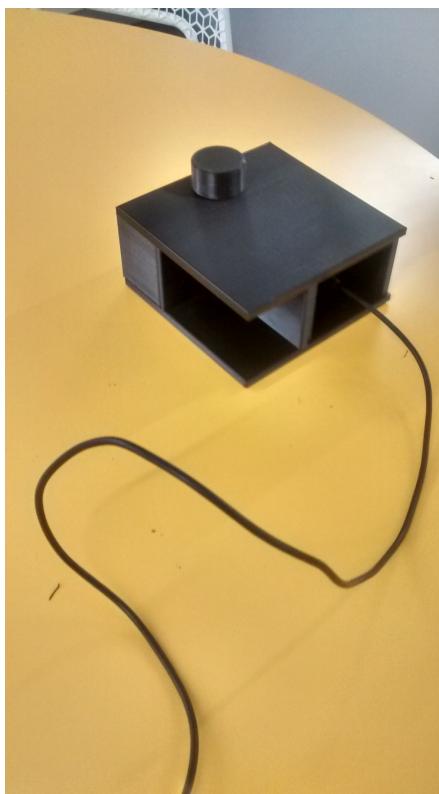
We used [Onshape](#) to design the files -- these files are available for copying / modification [here](#).



The resultant 3D STL files (suitable for 3D printing) are also available online [here](#).



We printed and assembled the 3D enclosure, including the protoboard with the C12880 module, and a USB cable for connecting to a laptop:

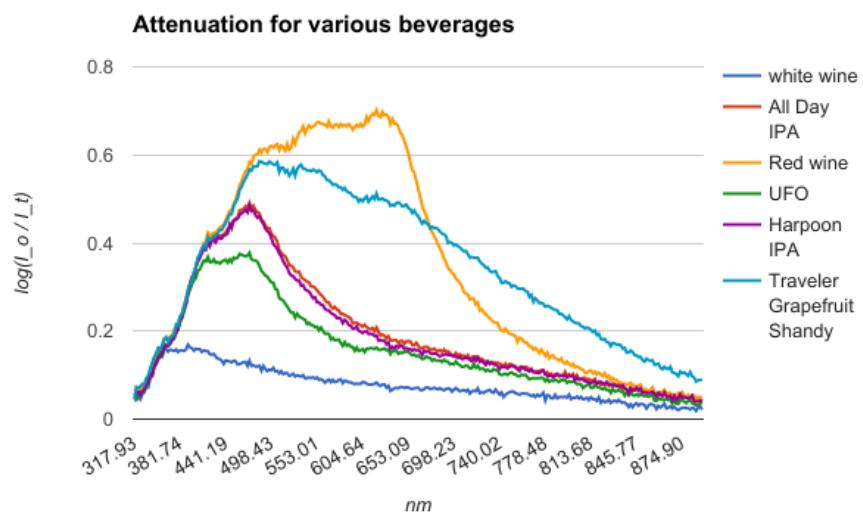


The enclosure also contains an inexpensive [Xenon bulb](#) as a light source:

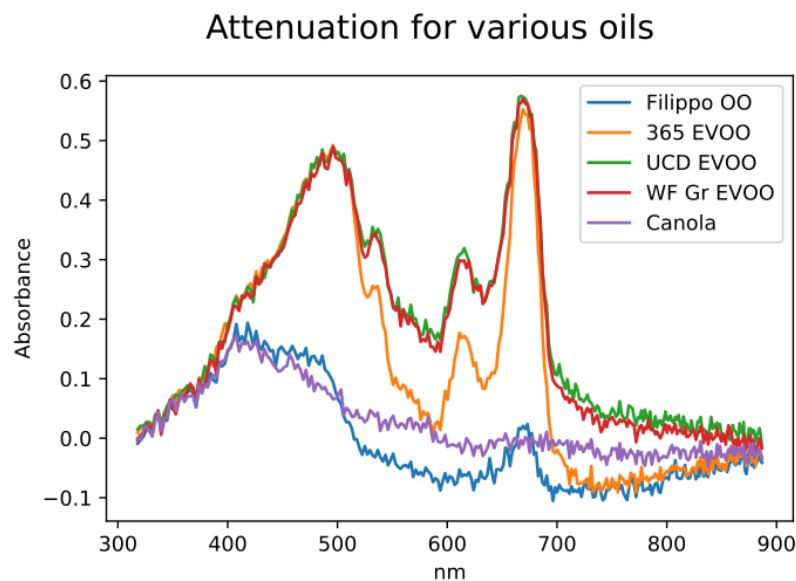


Basic Applications

A first test of our transmission spectroscopy setup using the C12880 involved testing vials of various alcoholic beverages. In this case, our 'blank' was a vial container Cambridge tap water.



Our final experiment involved assessing the attenuation of various oils, intending to compare various olive oils with canola oil:



Conclusion

The C12880MA is a convenient, easy to use spectroscopy module. It can be readily used with low-cost microcontrollers, and data can be obtained and analyzed using open source code running on a wide range of computer platforms.

