NP501-00019-2v0-E

FRAM
MB85RC256V

MB85RC256V is a 256K-bits FRAM with serial interface (I^2C), using the ferroelectric process and CMOS process technologies for forming the nonvolatile memory cells. Because FRAM is able to write high-speed even though a nonvolatile memory, it is suitable for the log management and the storage of the resume data, etc.

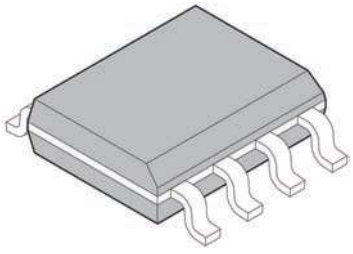
■ FEATURES

- **Bit configuration** : 32,768 words × 8 bits
- **Two-wire serial interface** : Fully controllable by two ports through serial clock (SCL) and serial data (SDA).
- **Operating frequency** : 1 MHz (Max.)
- **Read/write endurance** : 10^{12} times / byte
- **Data retention** : 10 years (+ 85 °C), 95 years (+ 55 °C), over 200 years (+ 35 °C)
- **Operating power supply voltage** : 2.7V to 5.5V
- **Low power consumption** : Operating current 200µA (Max. @1MHz),
  : Standby current 27µA (Typ.)
- **Operation ambient temperature range** : − 40 °C to + 85 °C
- **Package** : 8-pin plastic SOP (FPT-8P-M02)
  : 8-pin plastic SOP (FPT-8P-M08)
  RoHS compliant

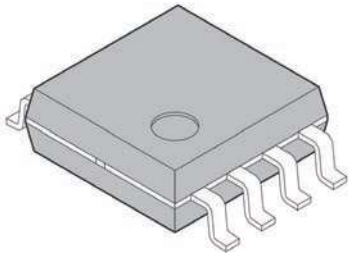
■ ORDERING INFORMATION

Product name	Package	Shipping form
MB85RC256VPNF-G-JNE1	8-pin plastic SOP (FPT-8P-M02) 3.90mm×5.05mm,1.27mm pitch	Tube
MB85RC256VPNF-G-JNERE1		Embossed Carrier tape
MB85RC256VPF-G-JNE2	8-pin plastic SOP (FPT-8P-M08) 5.30mm×5.24mm,1.27mm pitch	Tube
MB85RC256VPF-G-JNERE2		Embossed Carrier tape

■ PACKAGE EXAMPLE



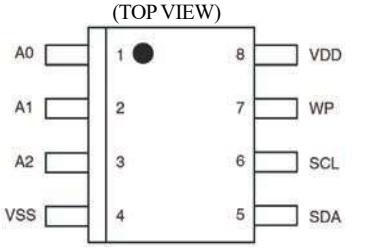
8-pin plastic SOP
(FPT-8P-M02)



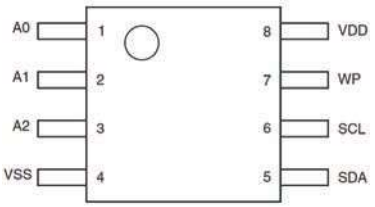
8-pin plastic SOP
(FPT-8P-M08)

■PIN ASSIGNMENT

(TOP VIEW)



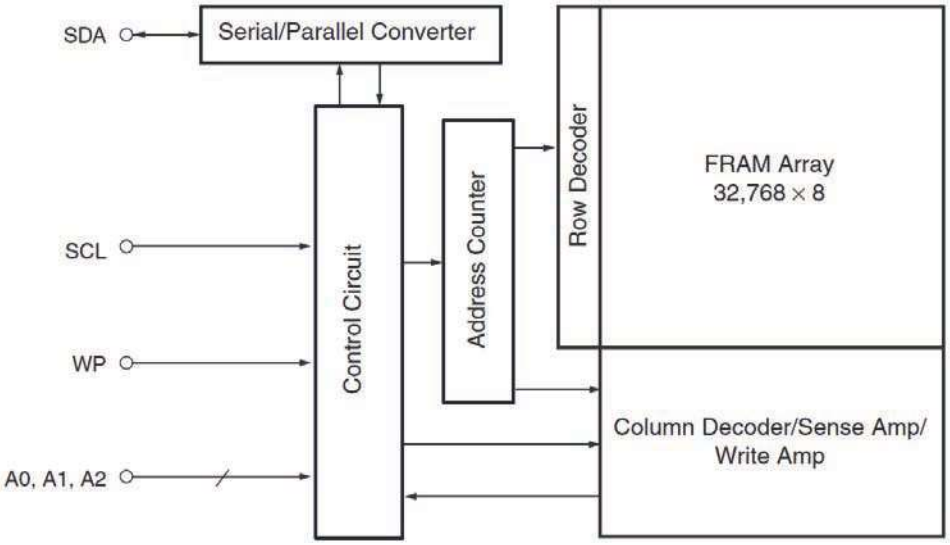
(FPT-8P-M02)



(FPT-8P-M08)

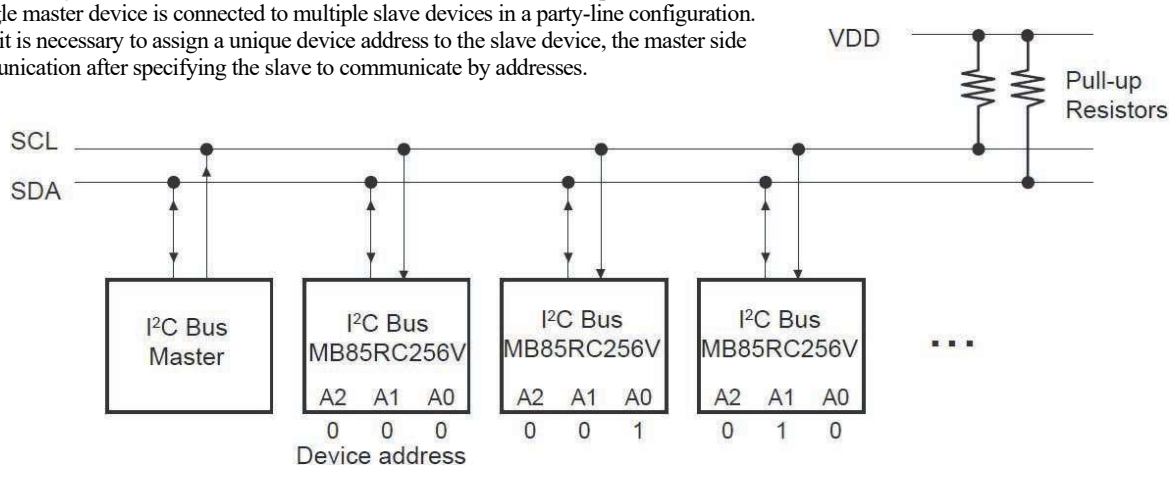
Pin No.	Pin name	Description
1 to 3	A0 to A2	Device Address pins MB85RC256V can be connected to the same data bus up to 8 devices. Device addresses are used in order to identify each of these devices. Connect these pins to VDD pin or VSS pin externally. Only if the combination of VDD and VSS pin matches a Device Address Code inputted from the SDA pin, the device operates. In the open pin state, A0, A1 and A2 are internally pulled-down and recognized as the "L" level.
4	VSS	Ground pin
5	SDA	Serial Data I/O pin This is an I/O pin which performs bidirectional communication for both memory address and writing/reading data. It is possible to connect multiple devices. It is an open drain output, so a pull-up resistor is required to be connected to the external circuit.
6	SCL	Serial Clock pin This is a clock input pin for input/output serial data. Data is sampled on the rising edge of the clock and output on the falling edge.
7	WP	Write Protect pin When Write Protect pin is the "H" level, writing operation is disabled. When Write Protect pin is the "L" level, the entire memory region can be overwritten. Reading operation is always enabled regardless of Write Protect pin input level. The Write Protect pin is internally pulled down to VSS pin, and that is recognized as the "L" level (write enabled) when the pin is the open state.
8	VDD	Supply Voltage pin

■BLOCK DIAGRAM



■I²C

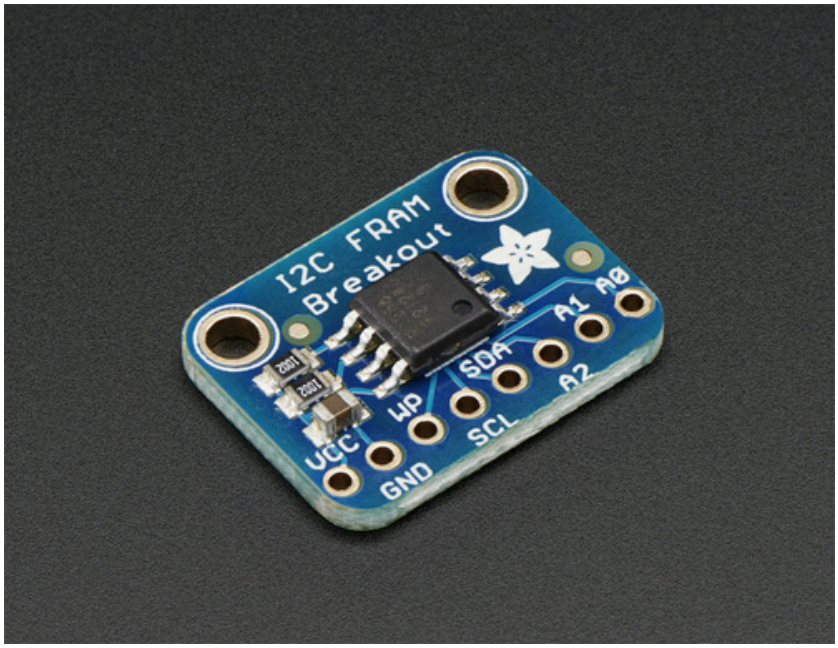
The MB85RC256V has the two-wire serial interface; the I²C bus, and operates as a slave device. The I²C bus defines communication roles of "master" and "slave" devices, with the master side holding the authority to initiate control. Furthermore, the I²C bus connection is possible where a single master device is connected to multiple slave devices in a party-line configuration. In this case, it is necessary to assign a unique device address to the slave device, the master side starts communication after specifying the slave to communicate by addresses.



adafruit learning system

Adafruit I2C FRAM Breakout

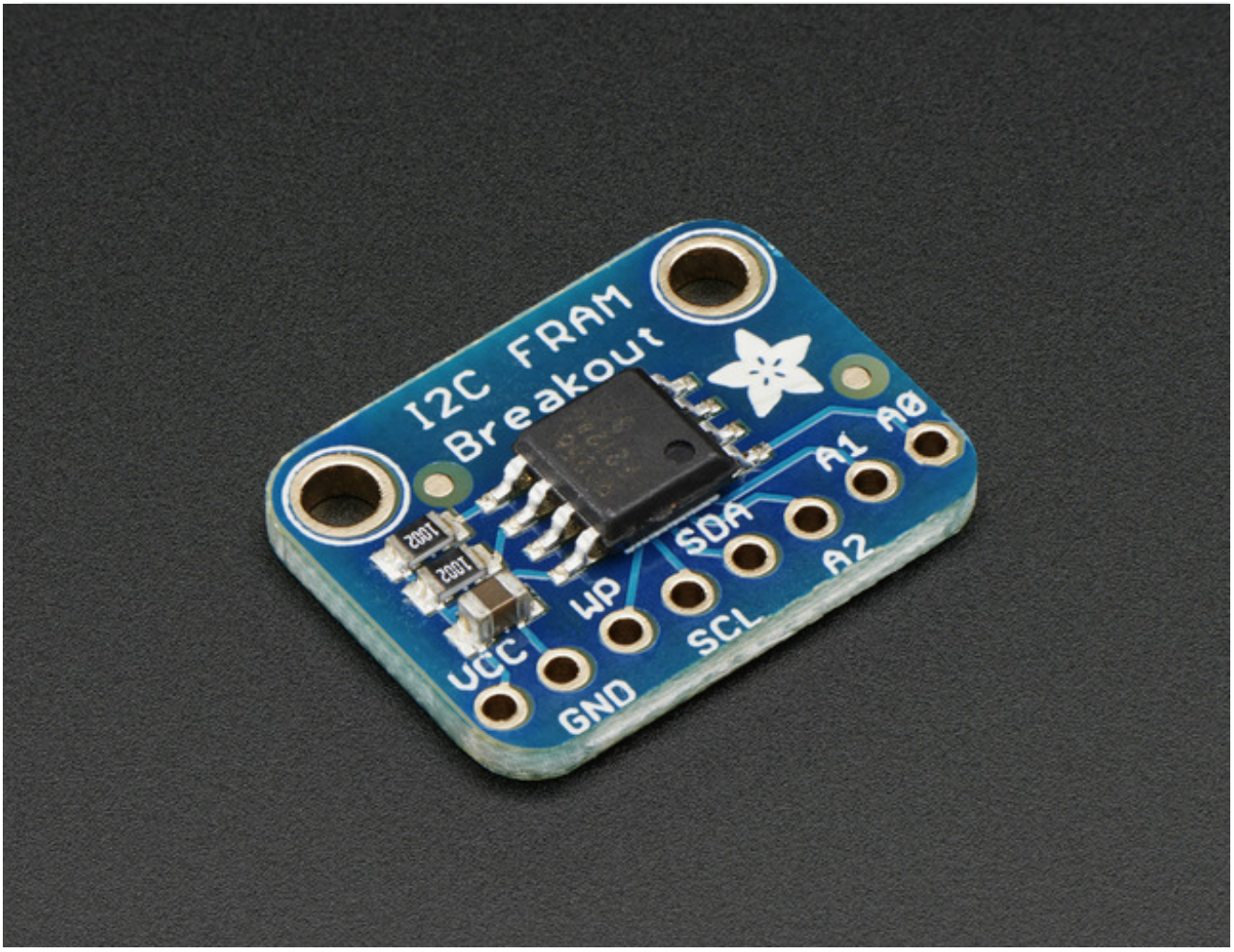
Created by lady ada



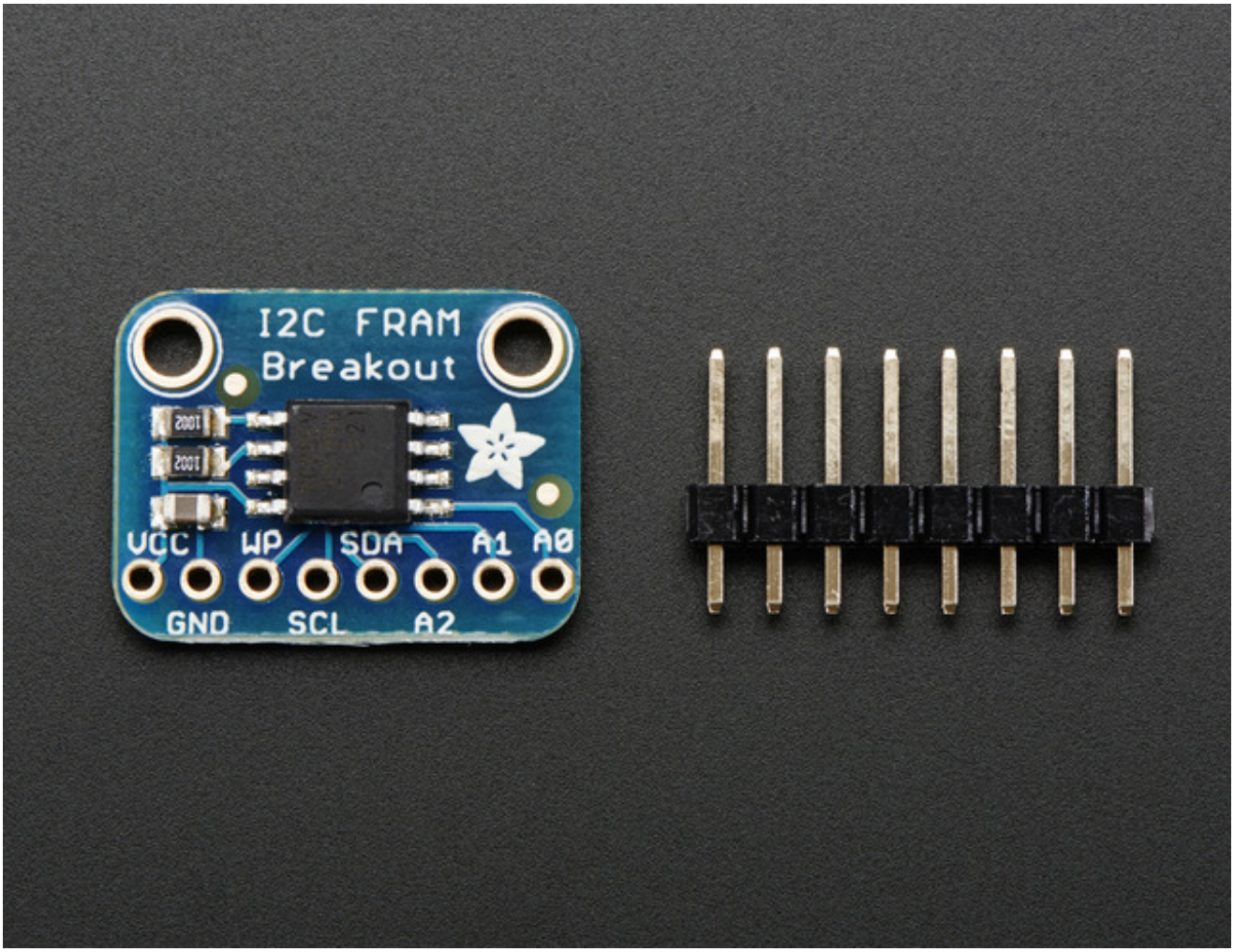
Last updated on 2014-06-28 08:15:15 PM EDT

Guide Contents

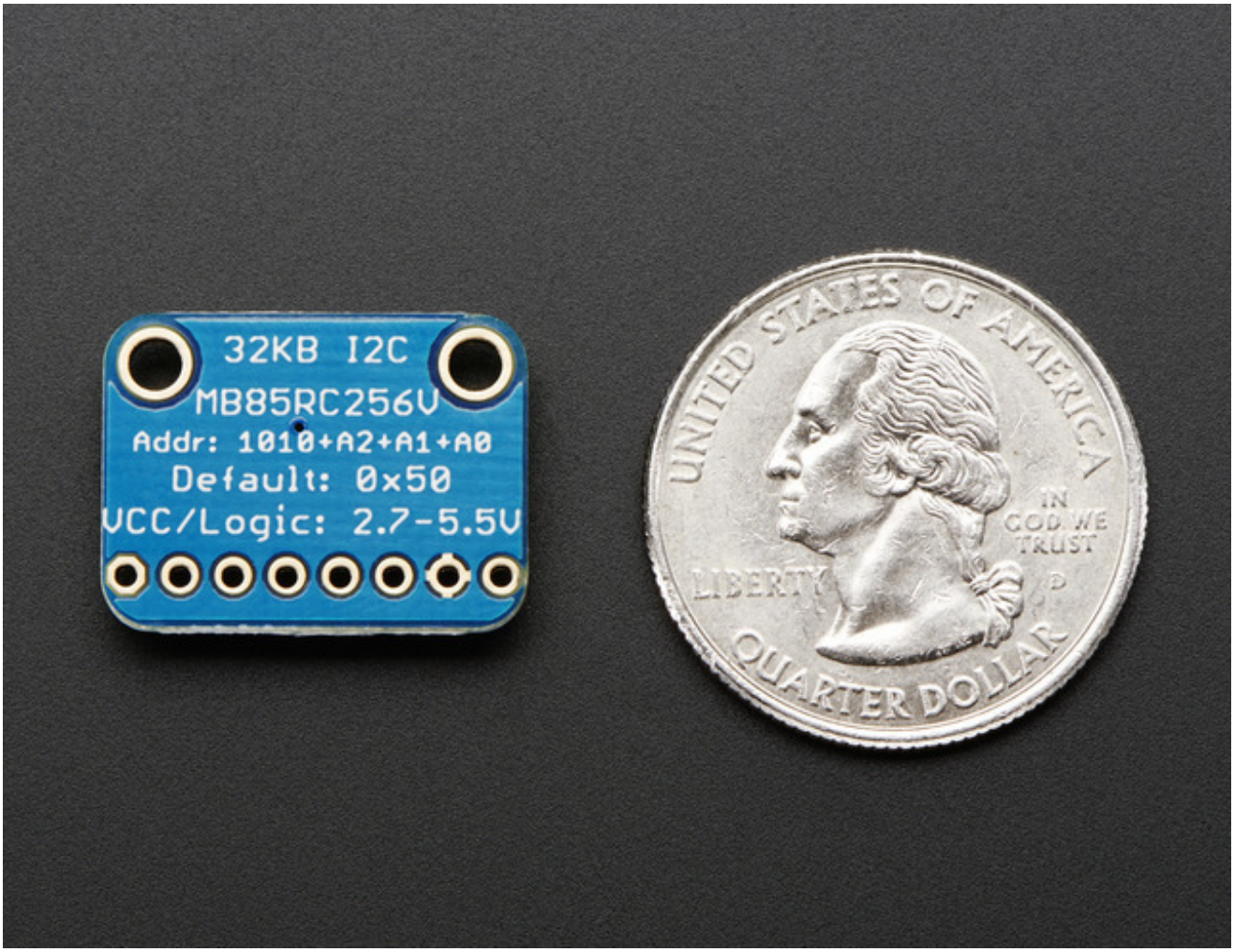
Overview



You're probably familiar with SRAM, DRAM, EEPROM and Flash but what about FRAM? FRAM is 'ferroelectric' RAM, which has some very interesting and useful properties. Unlike SRAM, FRAM does not lose the data when power is lost. In that sense it's a durable storage memory chip like Flash. However, it is much faster than Flash - and you don't have to deal with writing or erasing pages.

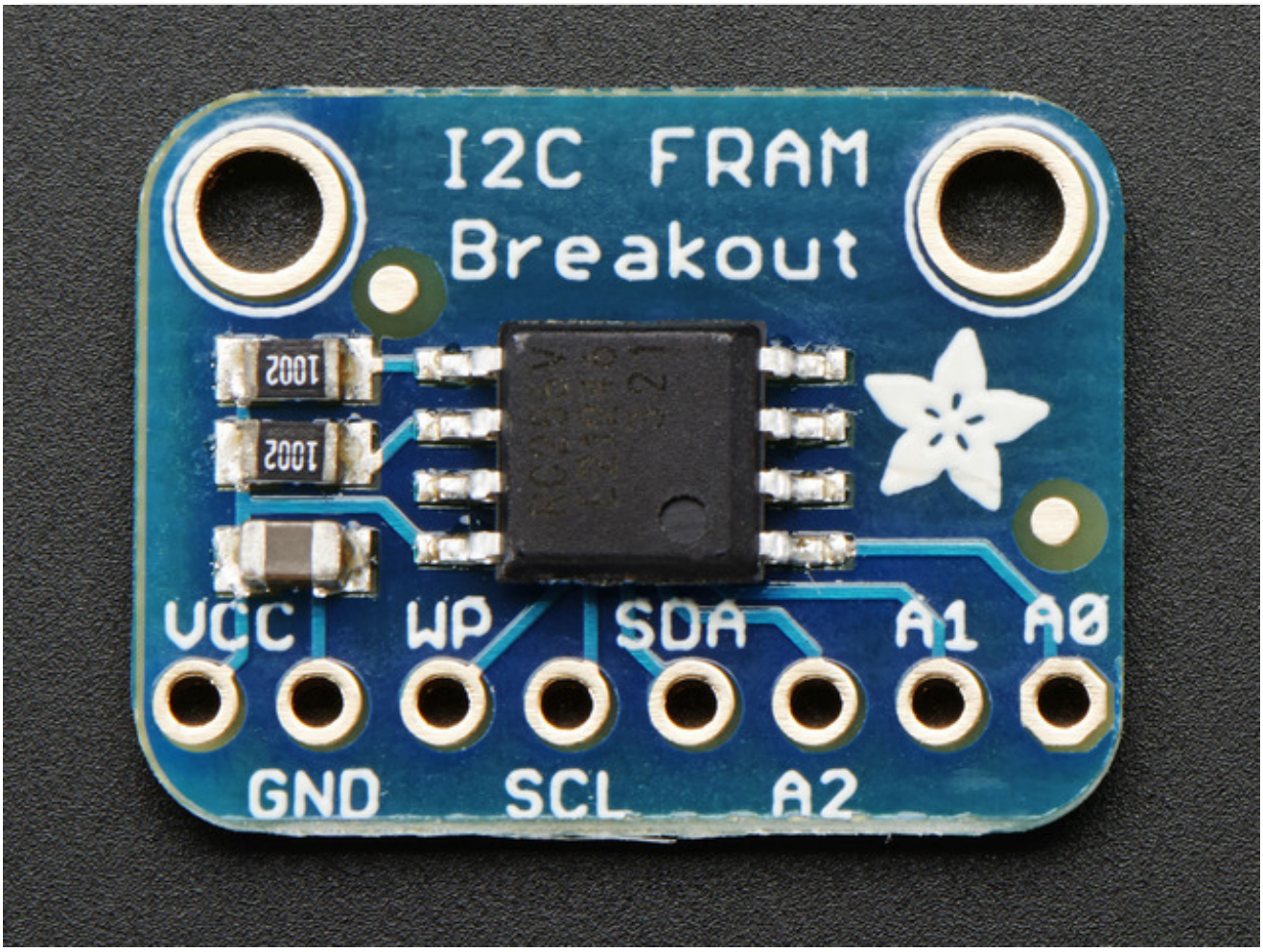


This particular FRAM chip has 256 Kbits (32 KBytes) of storage, interfaces using I2C, and can run at up to 1MHz I2C rates. Each byte can be read and written instantaneously (like SRAM) but will keep the memory for 95 years at room temperature. Each byte can be read/written 10,000,000,000,000 times so you don't have to worry too much about wear leveling.



With the best of SRAM and Flash combined, this chip can let you buffer fairly-high speed data without worrying about data-loss.

Pinouts



The FRAM chip is the little guy in the middle. On the bottom we have the power and interface pins

Power Pins:

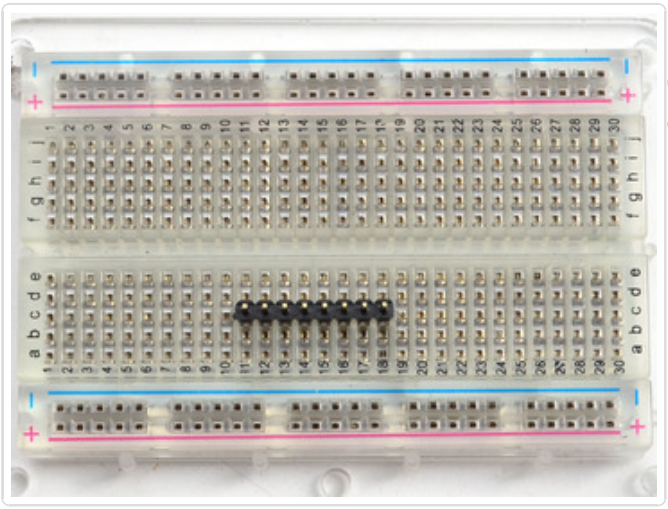
- **VCC** - this is the power pin. Since the chip uses 3-5VDC you should pick whatever the logic voltage you're using. For most Arduino's that's 5V.
- **GND** - common ground for power and logic

I2C Logic pins:

- **WP** - Write Protect pin. This is used to force write protection so you cannot write to the FRAM. It has an internal pulldown. Bring to a high voltage (VCC) to turn on WP
- **SCL** - I2C clock pin, connect to your microcontrollers I2C clock line.

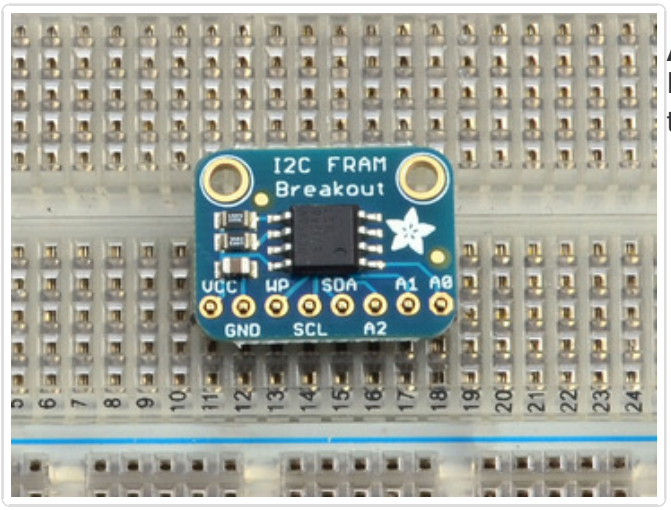
- **SDA** - I2C data pin, connect to your microcontrollers I2C data line.
- **A2, A1, A0** - These are the I2C address selection pins. By default the I2C address is 0x50. Connecting these pins to VCC and power cycling the chip will adjust the lower three bits of the address. For example, if A0 is high, the address is 0x51. If A1 and A2 are high, the address is 0x56

Assembly



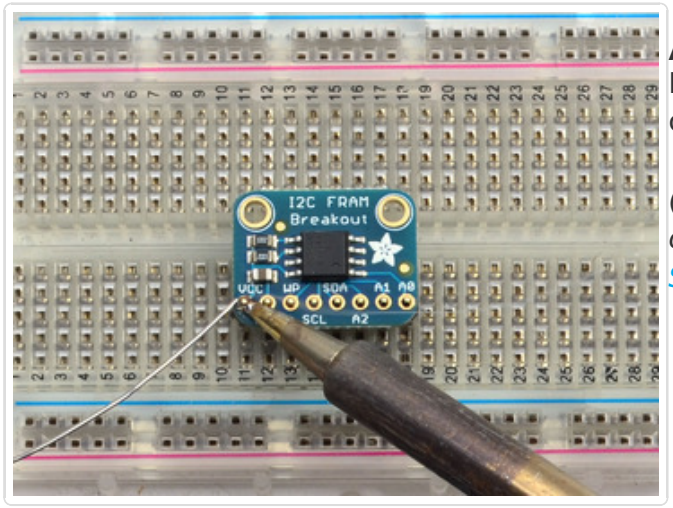
Prepare the header strip:

Cut the strip to length if necessary. It will be easier to solder if you insert it into a breadboard - **long pins down**



Add the breakout board:

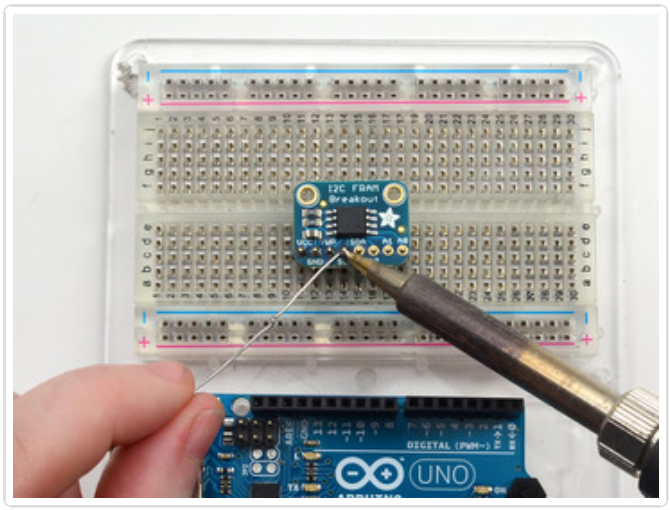
Place the breakout board over the pins so that the short pins poke through the breakout pads

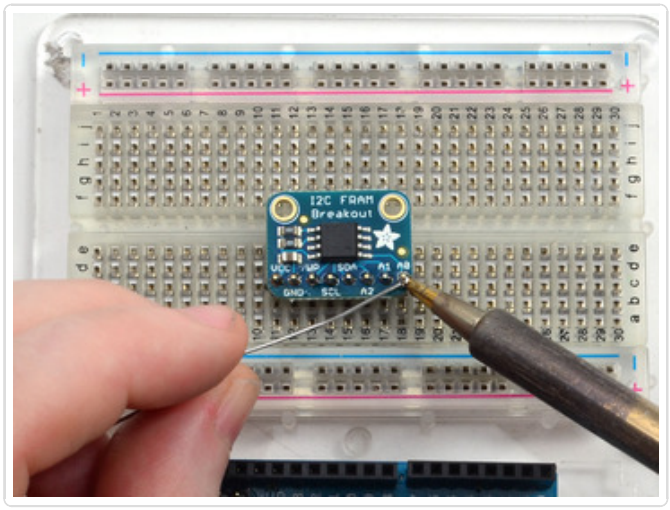


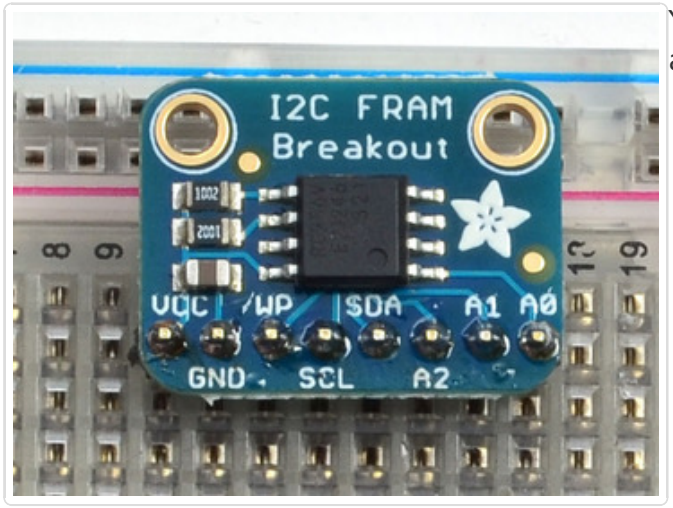
And Solder!

Be sure to solder all pins for reliable electrical contact.

(For tips on soldering, be sure to check out our Guide to Excellent Soldering (http://adafru.it/aTk)).

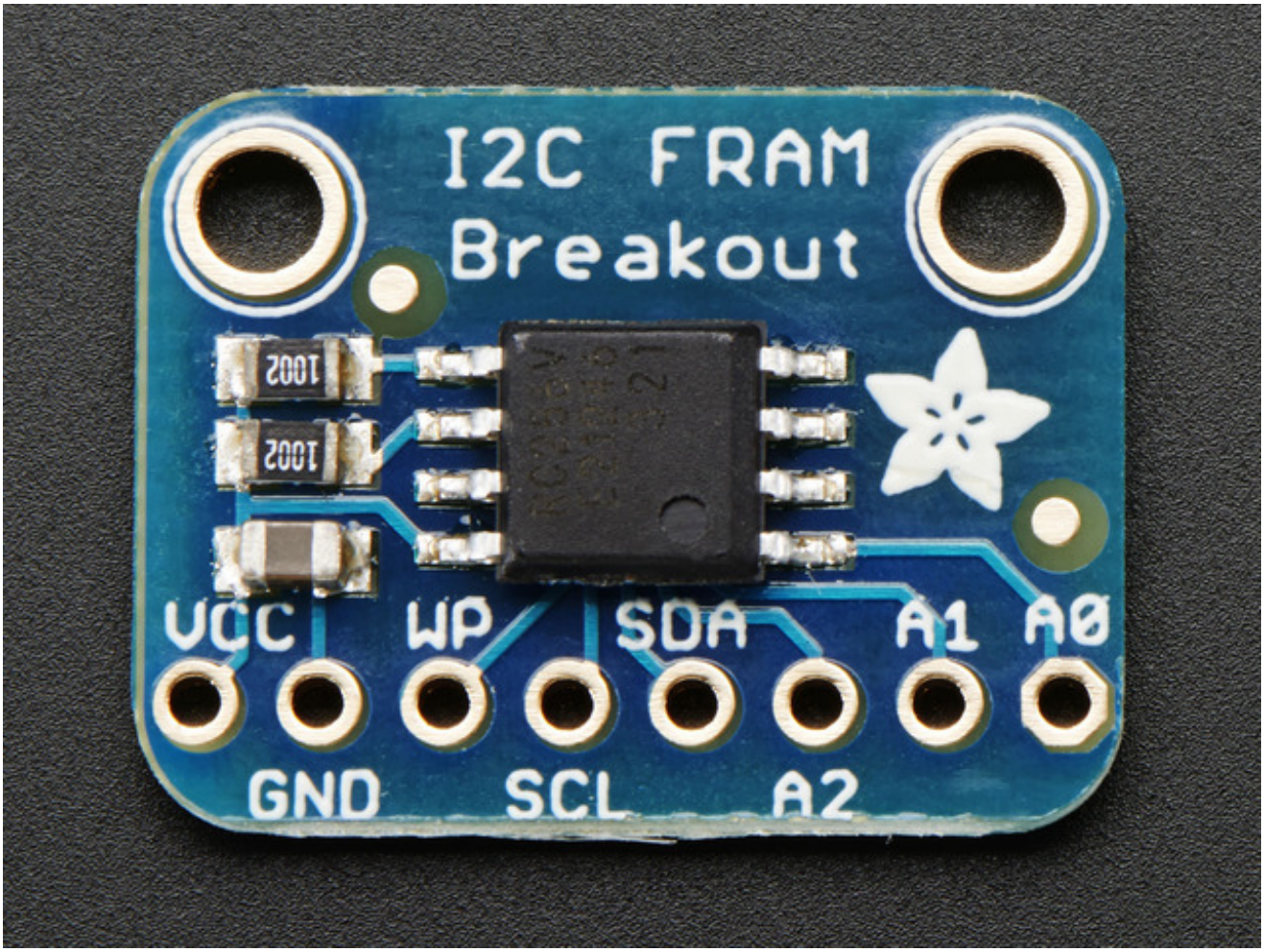




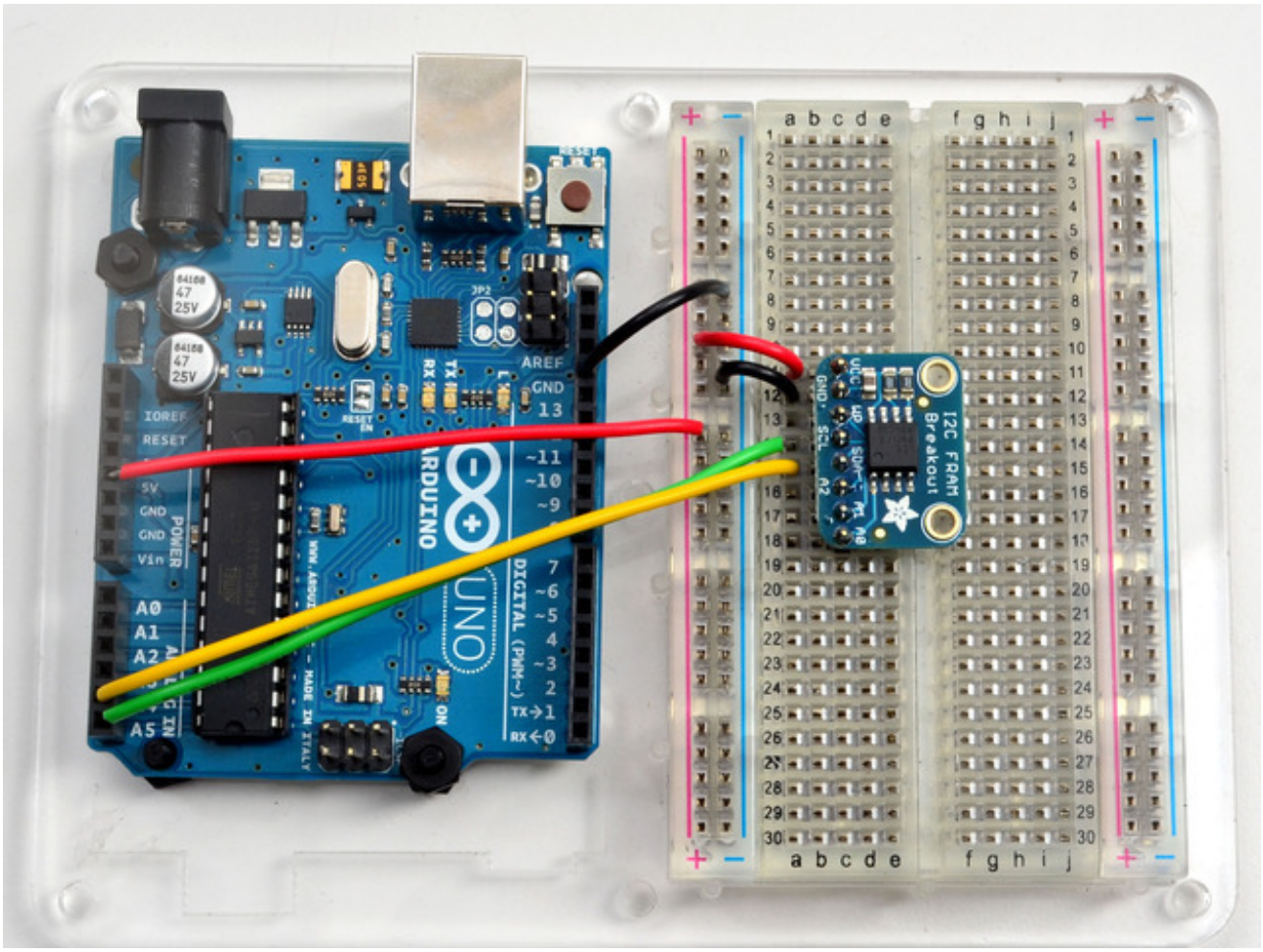


You're done! Check your solder joints visually and continue onto the next steps

Wiring and Test
Arduino Wiring



You can easily wire this breakout to any microcontroller, we'll be using an Arduino



- Connect **Vcc** to the power supply, 3V or 5V is fine. Use the same voltage that the microcontroller logic is based off of. For most Arduinos, that is 5V
- Connect **GND** to common power/data ground
- Connect the **SCL** pin to the I2C clock **SCL** pin on your Arduino. On an UNO & '328 based Arduino, this is also known as **A5**, on a Mega it is also known as **digital 21** and on a Leonardo/Micro, **digital 3**
- Connect the **SDA** pin to the I2C data **SDA** pin on your Arduino. On an UNO & '328 based Arduino, this is also known as **A4**, on a Mega it is also known as **digital 20** and on a Leonardo/Micro, **digital 2**

The I2C FRAM has a default I2C address of **0x50** but you can set the address to any of 8 values between 0x50 and 0x57 so you can have up to 8 of these chips all sharing the same SCL/SDA pins. We suggest just connecting up the power and SDA/SCL pins for now. Once you have things working, change up the address as desired.

Download Adafruit_FRAM_I2C

To begin reading and writing data to the chip, you will need to download Adafruit_FRAM_I2C from our github repository (http://adafru.it/dtq). You can do that by visiting the github repo and manually downloading or, easier, just click this button to download the zip

Download Adafruit FRAM i2c Library

http://adafru.it/dtr

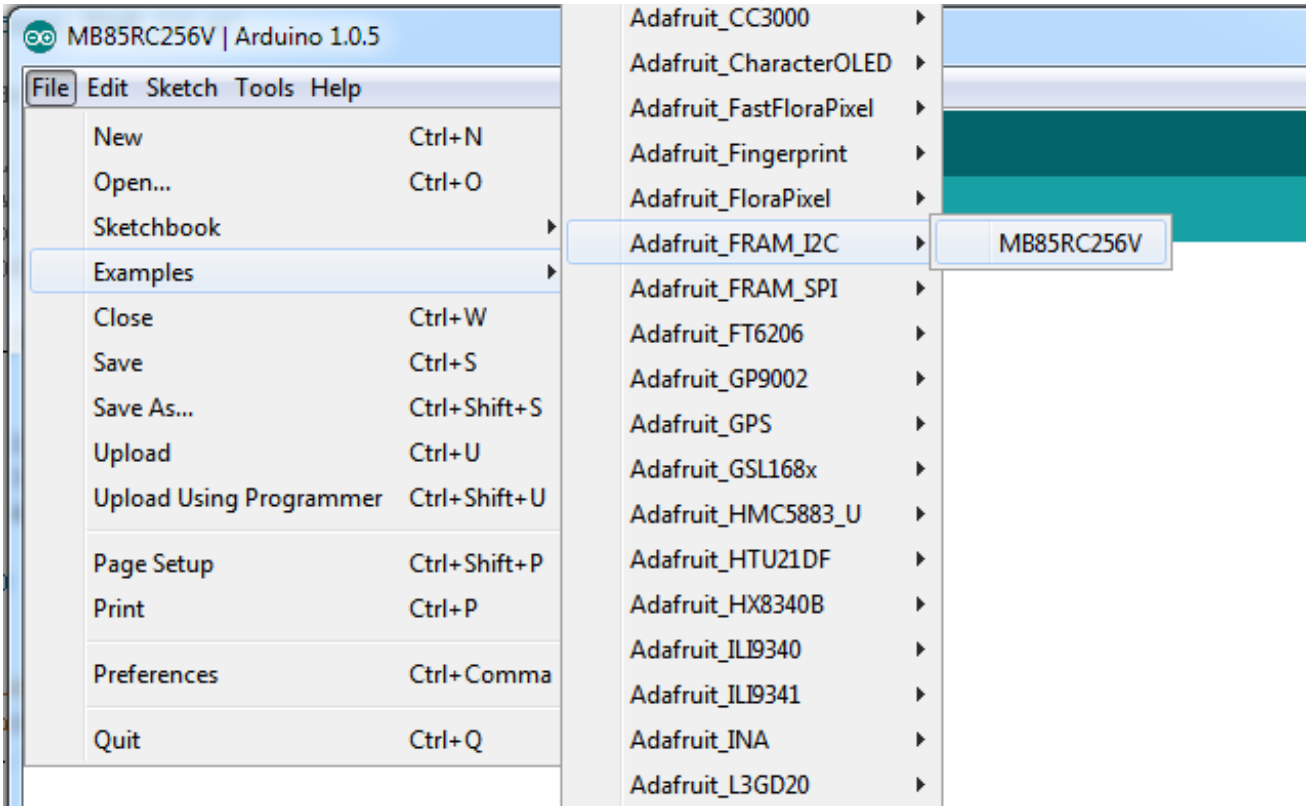
Rename the uncompressed folder **Adafruit_FRAM_I2C** and check that the **Adafruit_FRAM_I2C** folder contains **Adafruit_FRAM_I2C.cpp** and **Adafruit_FRAM_I2C.h**

Place the **Adafruit_FRAM_I2C** library folder your **arduinosketchfolder/libraries/** folder.
You may need to create the **libraries** subfolder if its your first library. Restart the IDE.

We also have a great tutorial on Arduino library installation at:
http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use (http://adafru.it/aYM)

Load Demo

Open up **File->Examples->Adafruit_FRAM_I2C->MB85RC256V** and upload to your Arduino wired up to the sensor

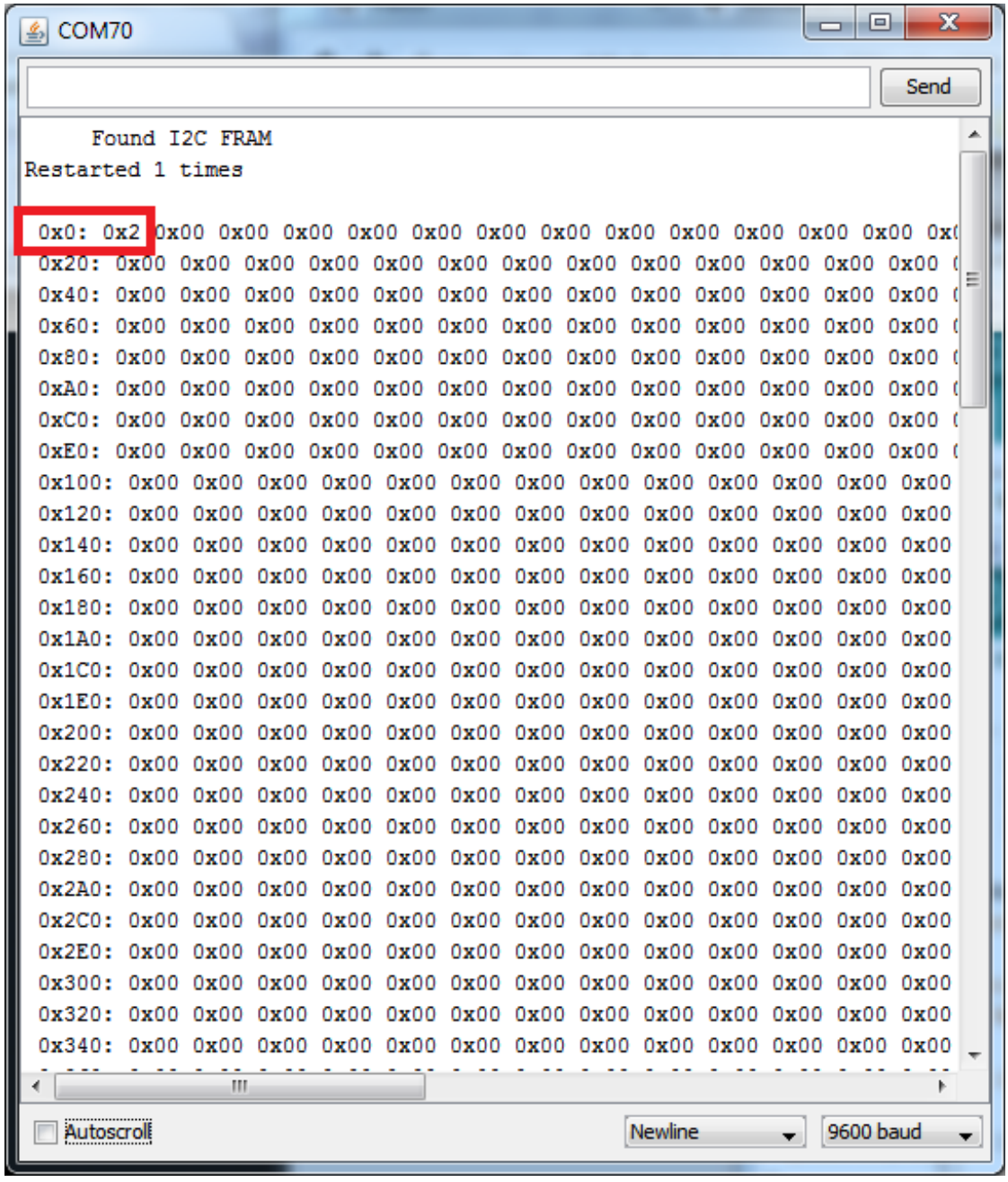


Thats it! Now open up the serial terminal window at 9600 speed to begin the test.

The test is fairly simple - It first verifies that the chip has been found. Then it reads the value

written to location #0 in the memory, prints that out and write that value + 1 back to location #0. This acts like a restart-meter: every time the board is reset the value goes up one so you can keep track of how many times its been restarted.

Afterwards, the Arduino prints out the value in every location (all 256KB!)



Library Reference

The library we have is simple and easy to use

You can create the FRAM object with

```
Adafruit_FRAM_I2C fram = Adafruit_FRAM_I2C();
```

Then when you **begin**(), pass in the i2c address. The default is 0x50 so if you don't put any value in the default is used.
If you have different addresses, call something like

```
fram.begin(0x53)
```

for example.

Then to write a value, call

```
fram.write8(address, byte-value);
```

to write an 8-bit value to the address location
Later on of course you can also read with

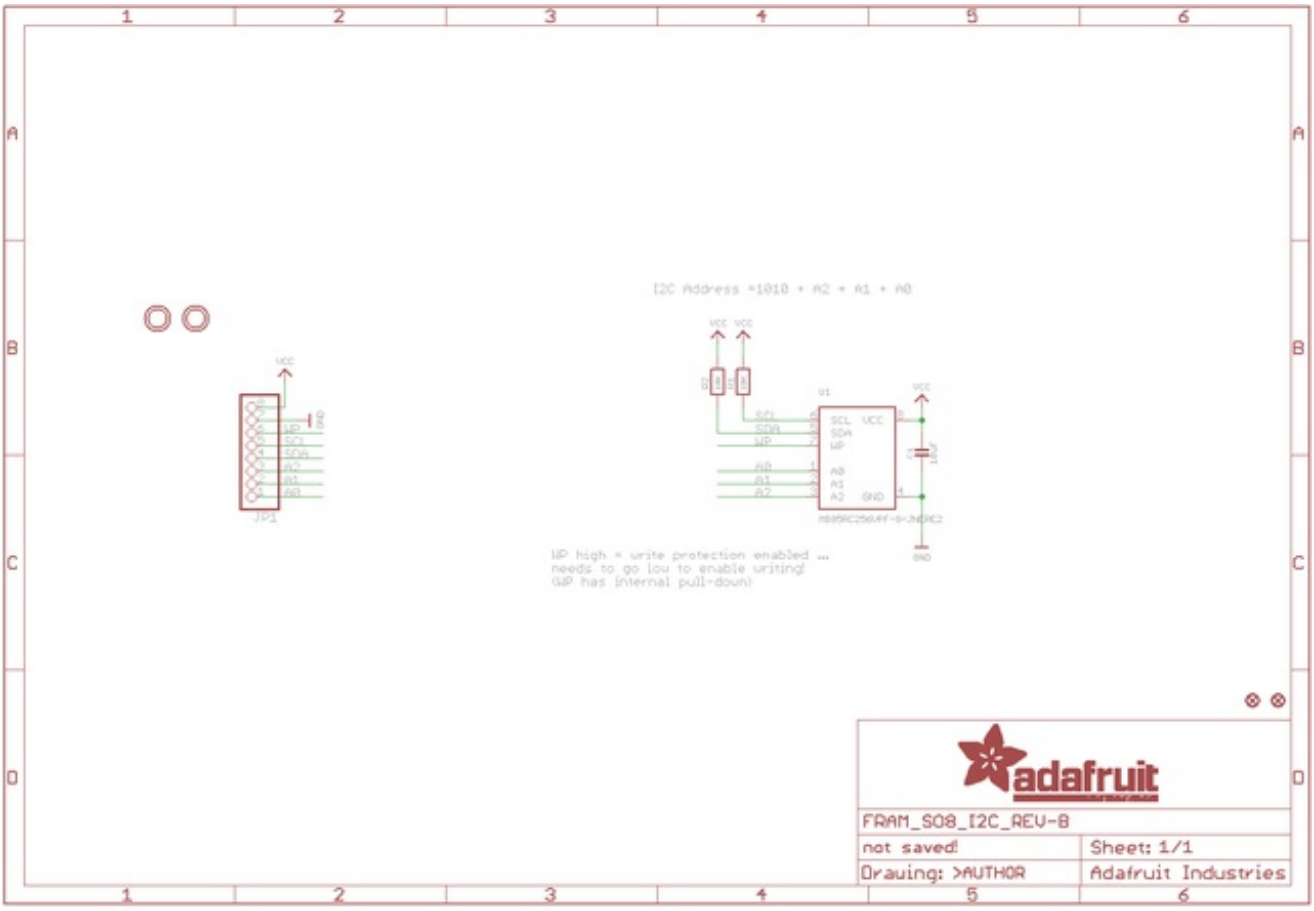
```
fram.read8(address);
```

which returns a byte reading.

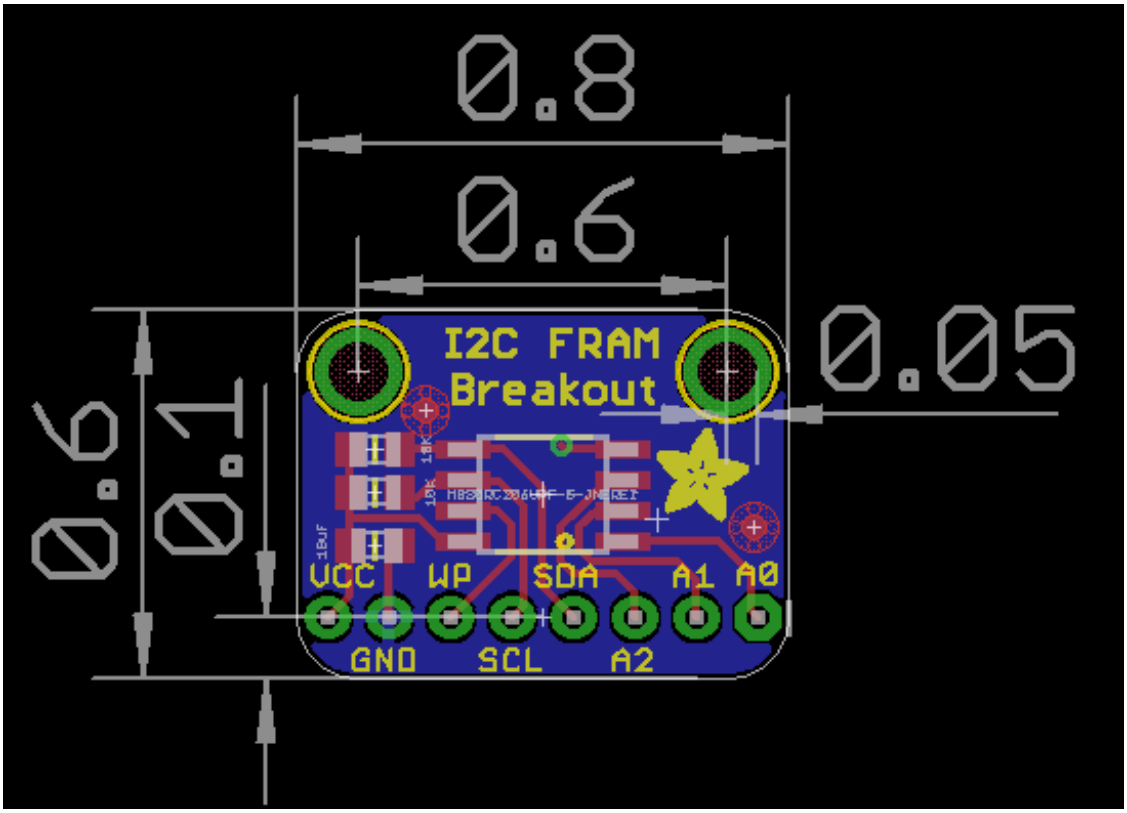
Downloads

Datasheet for the MB85RC256VPF FRAM chip (http://adafru.it/dt7)

Schematics



Fabrication Print


0.8
0.6
0.05
0.6
0.1
I2C FRAM
Breakout
VCC
WP
SDA
A1
A0
GND
SCL
A2