

# AMS Firmware Guide

## ECE 492 - Spring 2015

This document describes the I2C interface to the AMS firmware, including the provided functionality as well as how to test the boards outside of the pack. Also included is a brief overview of how the software operates. Any questions regarding the software functionality should be directed towards Alex Hytha at [Hythaam@gmail.com](mailto:Hythaam@gmail.com).

---

### AMS I2C Interface

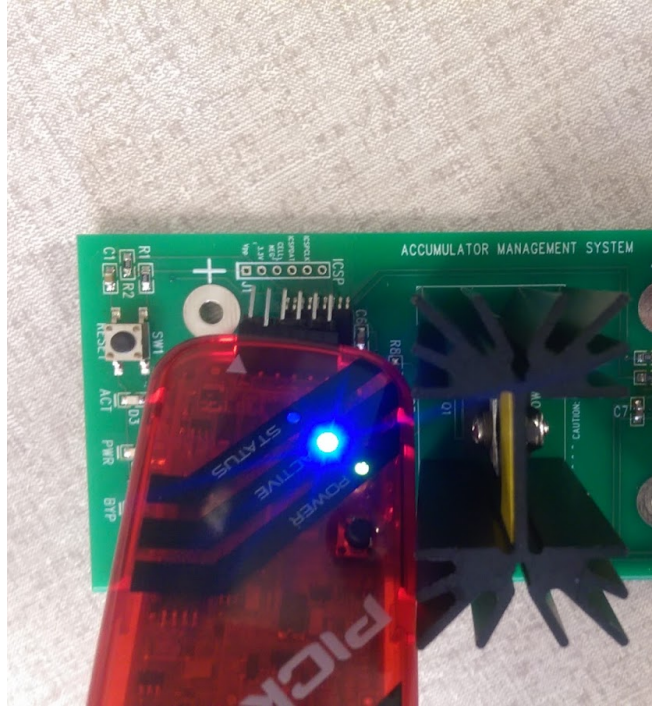
The following table describes the available functions in the current firmware revision. Included in the table is the command code as well as the I2C message structure:

Command	Code	Description
RESET_BY	0x00	Resets the bypass timer
SET_BY	0x01	Sets the device into bypass mode
PANIC	0x02	Rapidly blink LED - doesn't affect functionality
RELAX	0x03	End panic mode
SET_ADDR	0x04	Set device I2C address
SET_BYTIME	0x05	Set the bypass time

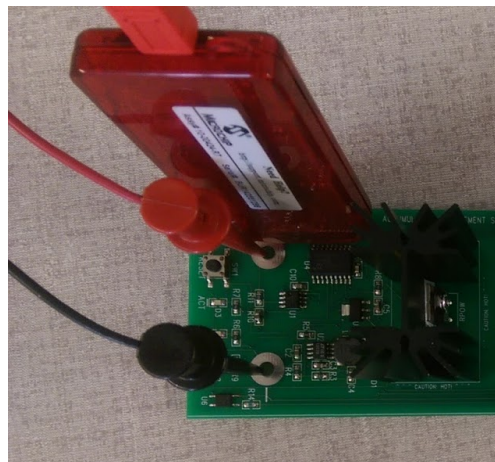
## Device Programming and Testing

To Program and test a device, you need a power supply capable of outputting at least 3 volts and 25 mA, a "PICKIT 3" programmer, and an "I2C2PC" adapter. A computer will be needed with the PICKIT 3 software and RealTerm installed.

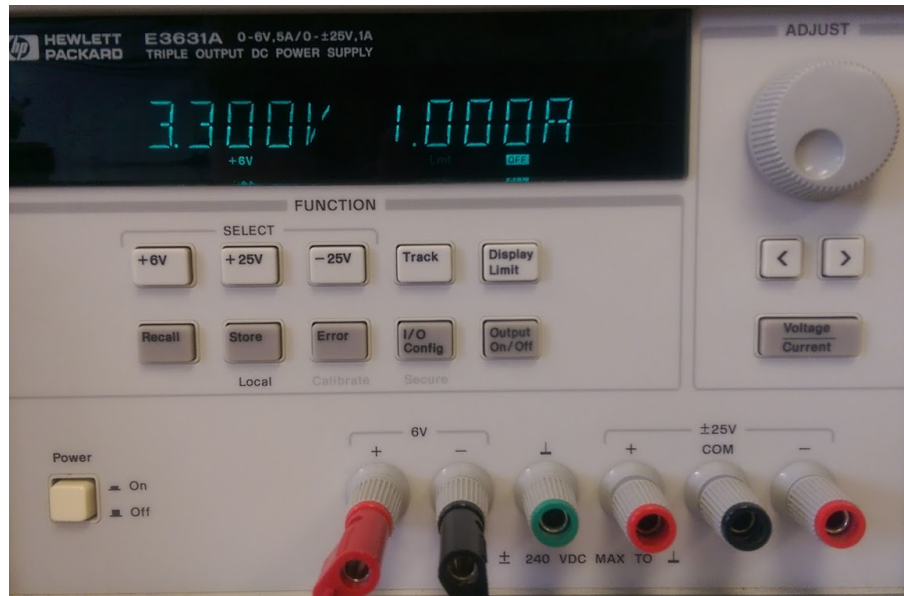
1. Attach the PICKIT 3 programmer to the AMS board.



2. Connect the AMS board to the power supply.



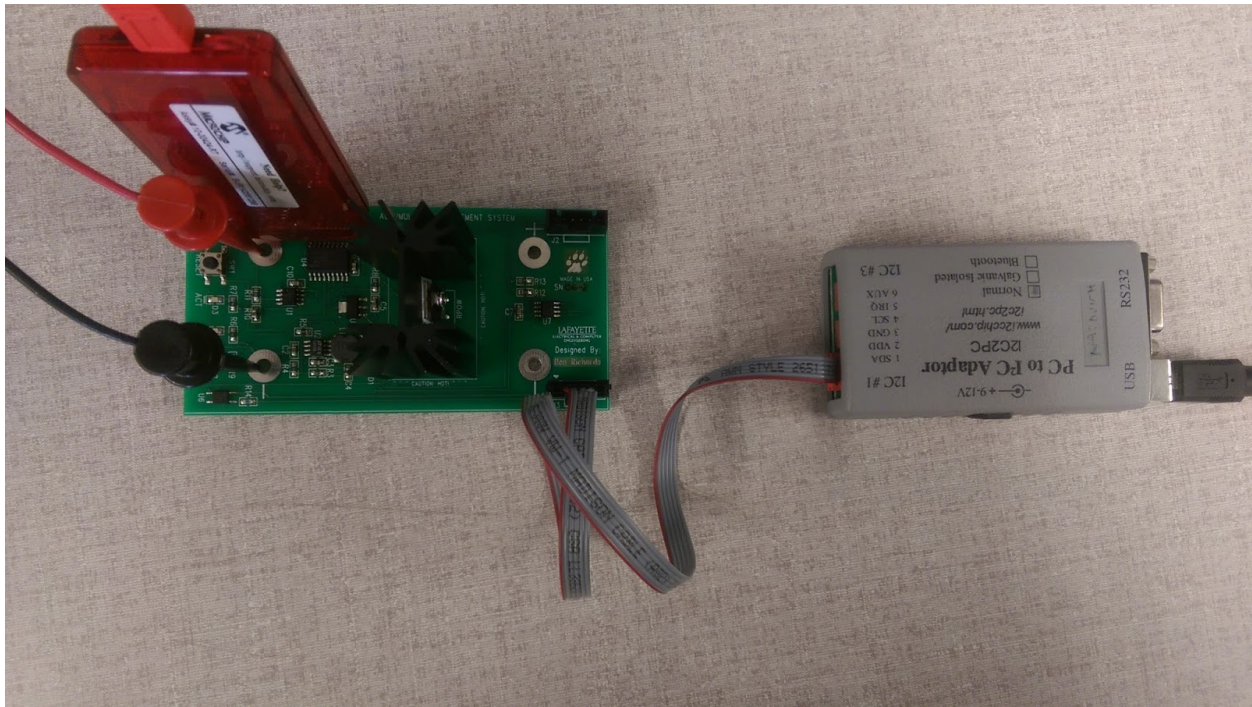
3. Set the power supply to 3.3 volts with a limit of 1 A.



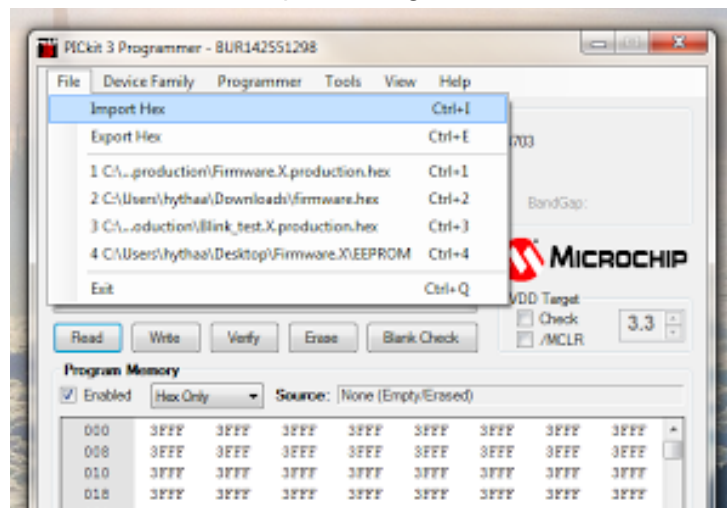
4. Enable power supply output. The voltage should settle to near 26mA.



5. Connect the I2C2PC adapter to the I2C bus on the AMS board.
  - a. **NOTE:** The cable has been know to have connection issues. It may be necessary to check the line voltage with an oscilloscope to ensure it can communicate with the bus.

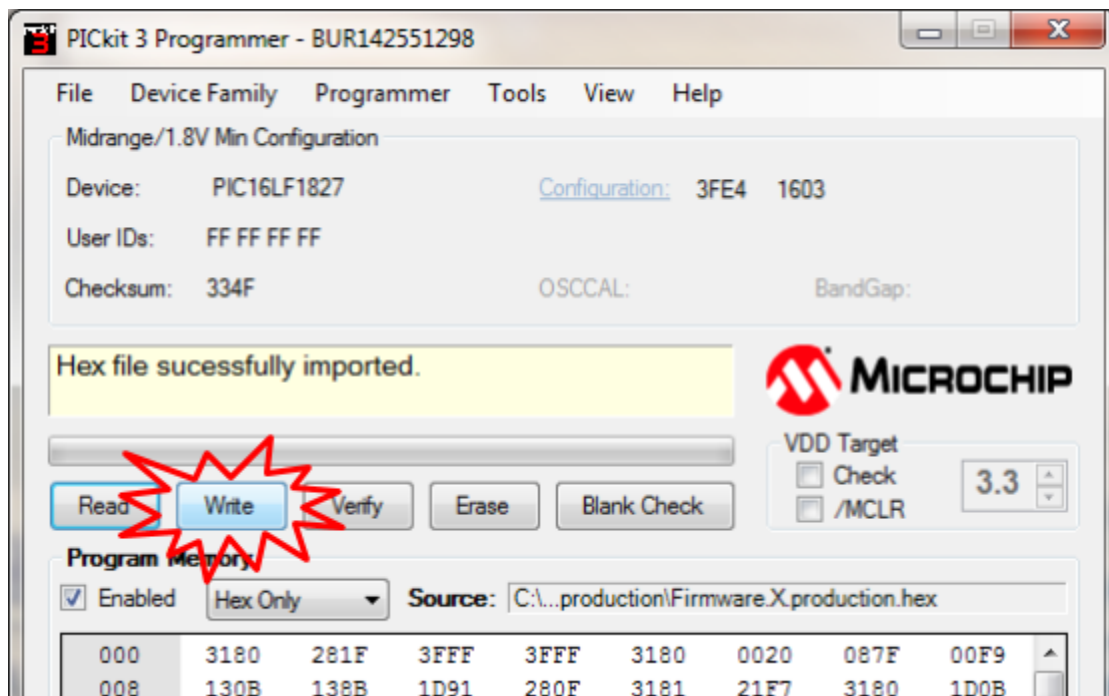


6. Start PICKIT 3.
  - a. Select the file from the import dialog.



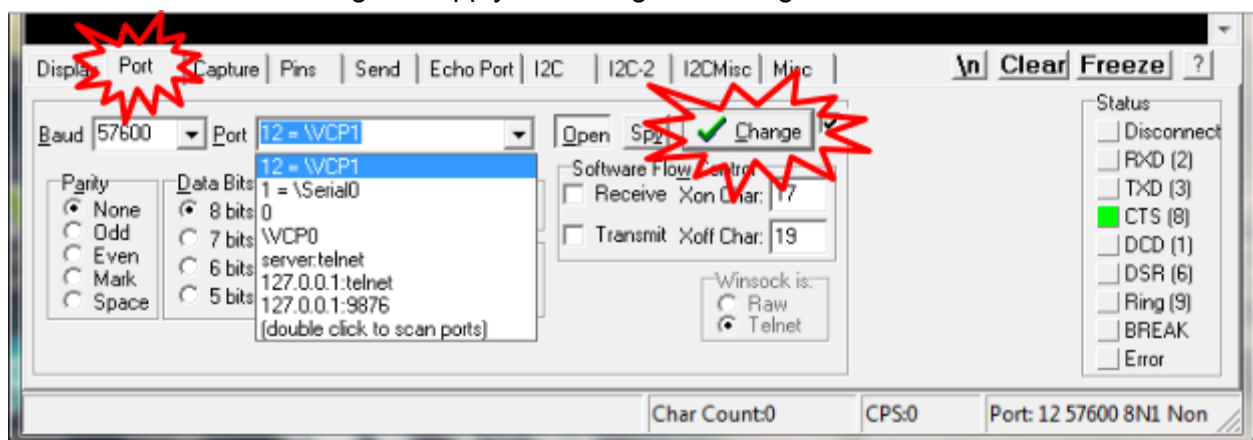


- b. Once the file has been imported, write it to the device.

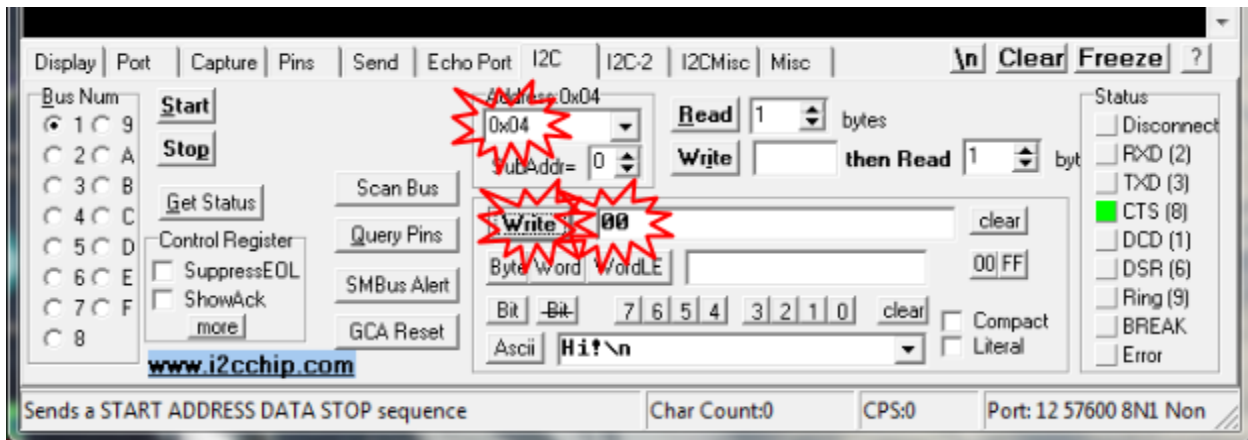


7. Start RealTerm.

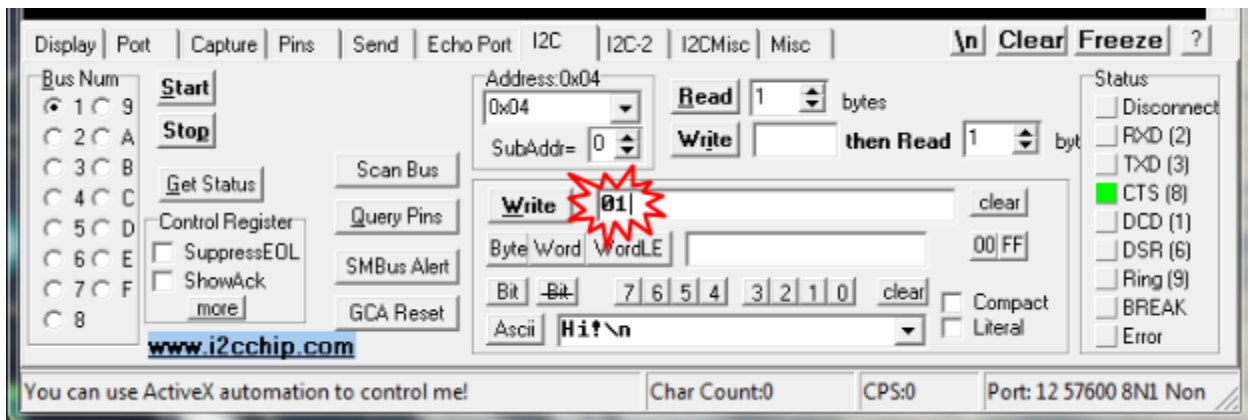
- a. In the "Port" tab, select the I2C2PC adapter as the port (port 12, in this case).  
b. Click "Change" to apply the change of settings.



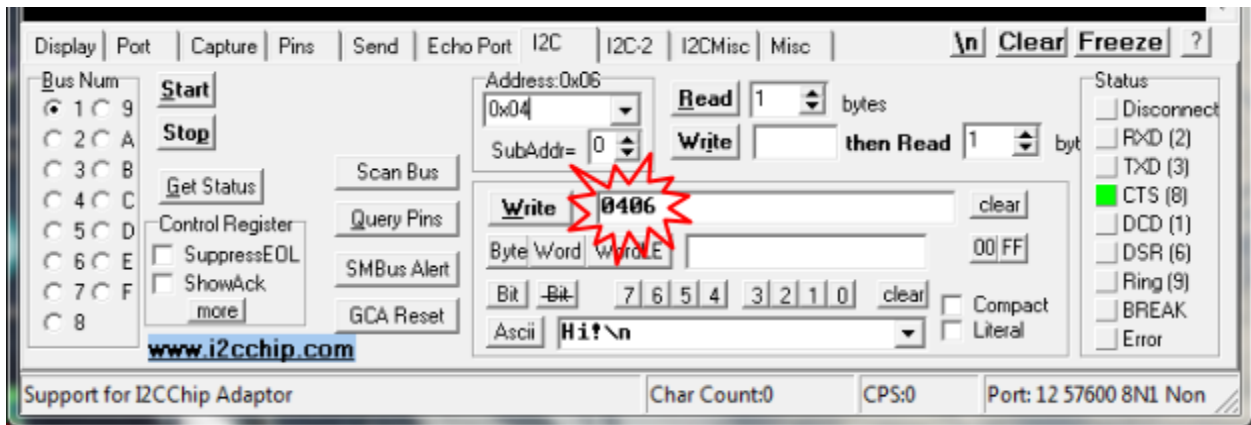
8. Test basic I2C functionality by sending byte 0x00 to address 0x04. The "ACT" LED should blink once when the command is received.



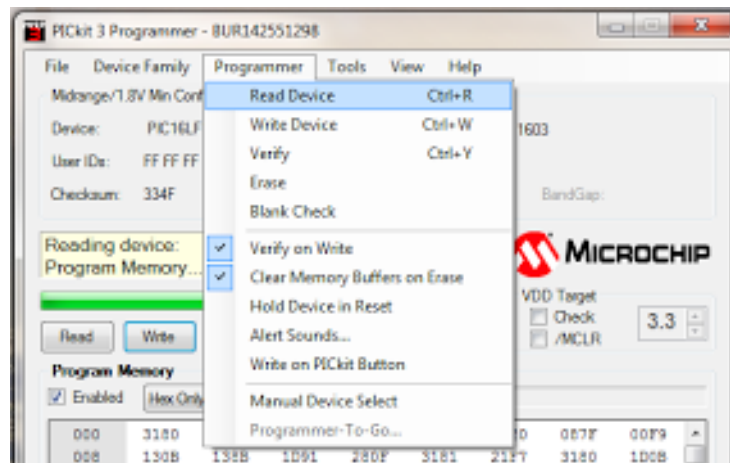
9. To test bypass mode, first increase the current limit of the power supply to 3A. This time, send the packet 0x01.



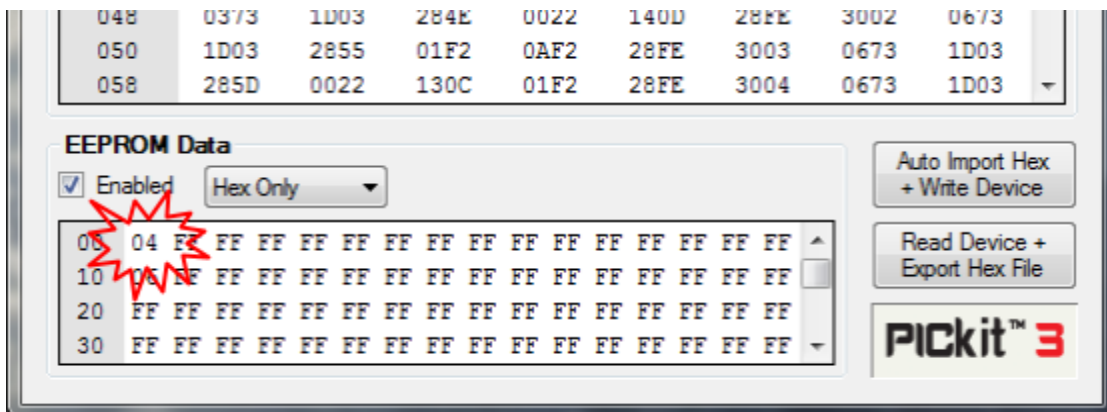
- a. The red "BYP" LED should light up and the current draw should increase to 2.9A.
  - b. Resend packet 0x00 or reset the device to end bypass mode
10. Test the SET\_ADDR command by sending the byte 0x04 followed by the new I2C address (here, it is 0x06) to change the address.



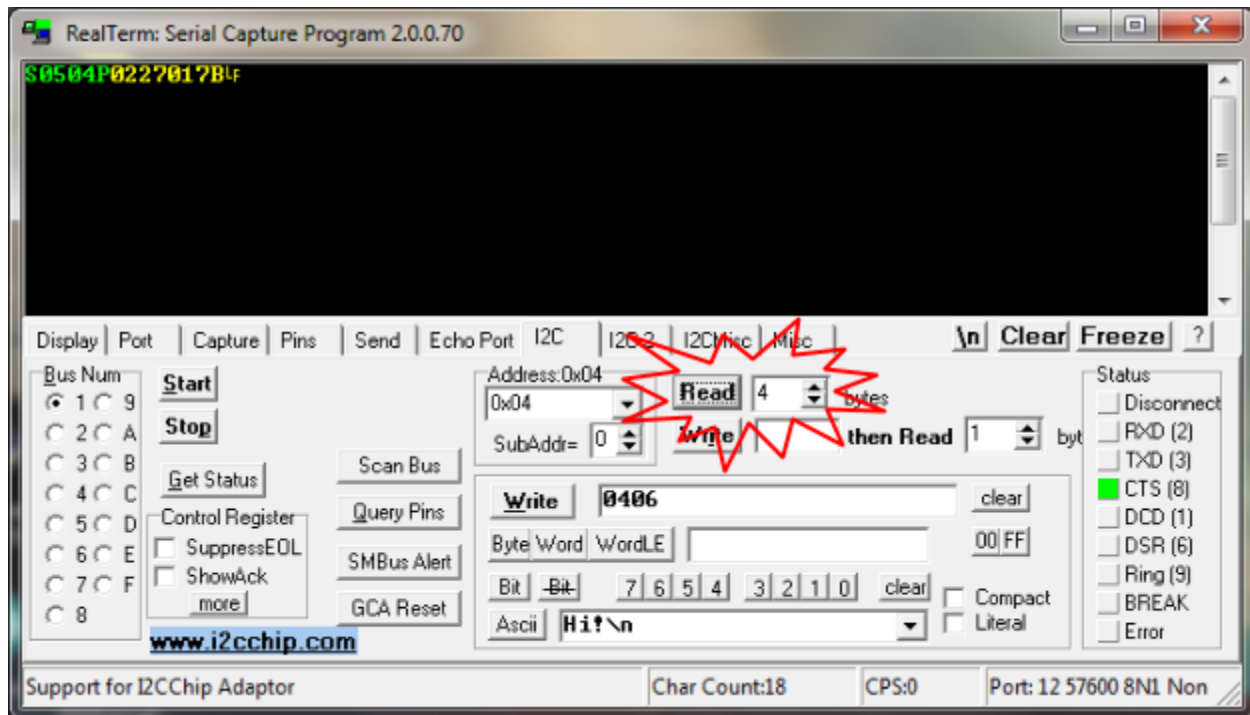
a. Confirm the change in address by reading the EEPROM data in PICKIT 3



b. The data stored in location 0x00 is the I2C address



11. Test the data retrieval by reading back 4 bytes from the device. The returned data should reflect different voltage values if the supply voltage is changed between iterations of this test.





## API documentation

### **RESET\_BY:** 0x00

This command resets the bypass timer within the device. If the bypass timer is not reset before it counts up to the bypass limit, the device will begin discharging the battery in bypass mode.

### **SET\_BY:** 0x01

This command directly sets the device into bypass mode.

### **PANIC:** 0x02

This command sets the panic flag in the device. The panic flag does nothing more than blink the “ACT” LED rapidly. None of the device functionality is affected.

### **RELAX:** 0x03

This command disables the panic flag.

### **SET\_ADDR:** 0x04

This command sets the I2C address. The command word MUST be followed by another byte to be set as the address. If this byte is not sent, the device can hang.

### **SET\_BYTIME:** 0x05

This command sets the bypass time limit for the device. The command word must be followed by 2 bytes to act as the bypass time limit. These bytes must be sent MSB first. The bypass time limit will be set to the given value multiplied by 32.8ms.

### **Reading data from the device:**

To retrieve data from the device, an I2C read must be initiated for exactly 4 bytes. If fewer or more bytes are read, the device can hang. The device will return two 16 bit values representing the voltage and temperature respectively. The values will be transmitted MSB first.