HP PRIME

Environmental Sensing

Mark Power
With thanks to contributors on MoHPC

HP PRIME TO USB DEVICES

- Recent HP Prime firmware includes:
- USBOpen() returns a list of connected USB-HID devices
- USBOpen(vid, pid) opens a specific device and returns a list of the output and input report lengths
- USBSend({data}) sends one USB packet of data
- USBRecv() returns a list of data items

HP PRIME TO USB TO I2C

- FT260 is a USB-HID device providing I2C and UART
- Default I2C configuration is usable as-is
- Default UART config is 19200 N81, so HP50g perhaps
- Prime limitations mean it cannot change FT260 config
- FT260EV1 from DigiKey US $15.21, Farnell UK £12.66

BOSCH BME680

- Low-power gas, temperature, barometric pressure and humidity sensor
- Uses include: indoor air quality monitoring, weather forecasting, temperature and humidity alerts
- 3.0mm x 3.0mm x 0.93mm
- Piminori PIM357 from DigiKey US $21.90, Farnell UK £16.40
- 53-page manual including pseudocode, register and configuration parameter tables
- Github repository with sample C code which shows that the manual is incomplete (no mention of signed/unsigned integers plus registers missing in the manual)

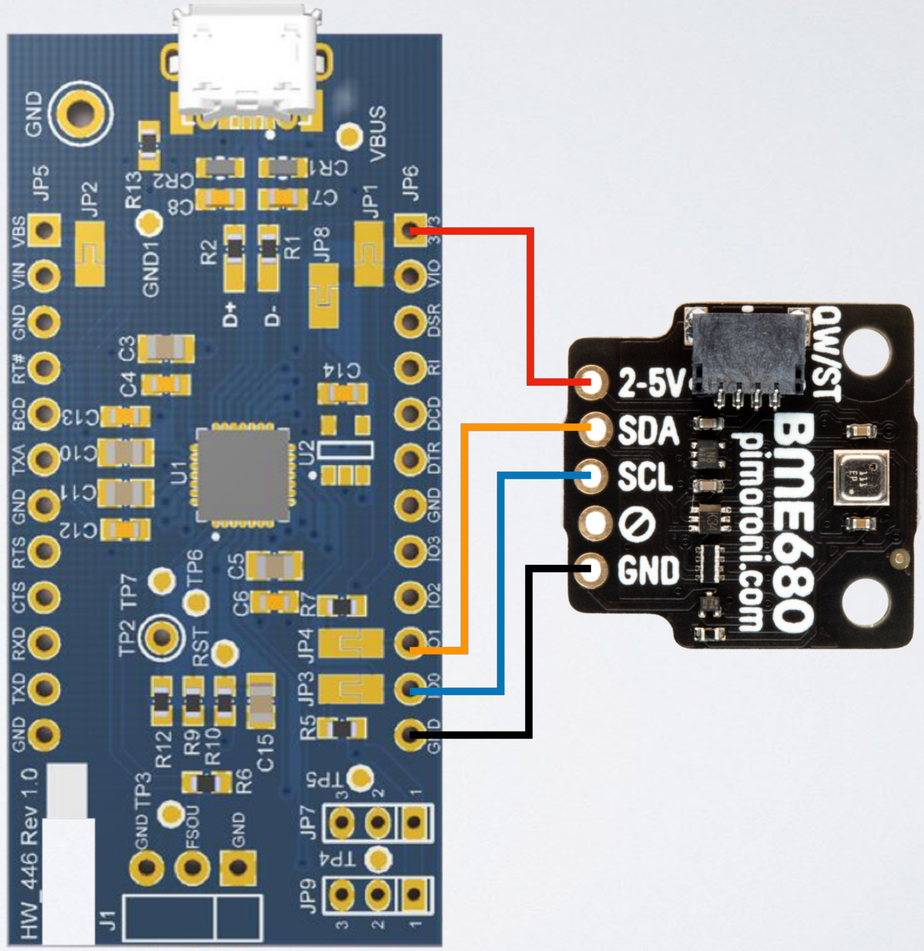
Mark's register map	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x	res_heat_val		<5:4> res_heat_range		range_switching_error											
1x														<7> new_data_0 <6> gas_measuring <5> measuring <3:0> gas_meas_index_0		press_adc MSB
2x	press_adc LSB	<7:4> press_adc XLSB	temp_adc MSB	temp_adc LSB	<7:4> temp_adc XLSB	hum_adc MSB	hum_adc LSB				gas_adc MSB	<7:6> gas_adc LSB <5> gas_valid_r <4> heat_stab_r <3:0> gas_range				
3x																
4x																
5x	idac_heat_0	idac_heat_1	idac_heat_2	idac_heat_3	idac_heat_4	idac_heat_5	idac_heat_6	idac_heat_7	idac_heat_8	idac_heat_9	res_heat_0	res_heat_1	res_heat_2	res_heat_3	res_heat_4	res_heat_5
6x	res_heat_6	res_heat_7	res_heat_8	res_heat_9	gas_wait_0	gas_wait_1	gas_wait_2	gas_wait_3	gas_wait_4	gas_wait_5	gas_wait_6	gas_wait_7	gas_wait_8	gas_wait_9		
7x	<3> heat_off	<4> run_gas <3:0> nb_conv	<6> spi_3w_int_en <2:0> osrs_h	<4> spi_mem_page	<7:5> osrs_t <4:2> osrs_p <1:0> mode	<4:2> filter <1> spi_3w_en										
8x											par_t2 LSB	par_t2 MSB	par_t3		par_p1 LSB	par_p1 MSB
9x	par_p2 LSB	par_p2 MSB	par_p3		par_p4 LSB	par_p4 MSB	par_p5 LSB	par_p5 MSB	par_p7	par_p6			par_p8 LSB	par_p8 MSB	par_p9 LSB	par_p9 MSB
Ax	par_p10															
Bx																
Cx																
Dx	chip_id (61h)															
Ex	reset	par_h2 MSB	<3:0> par_h1 LSB <7:4> par_h2 LSB	par_h1 MSB	par_h3	par_h4	par_h5	par_h6	par_h7	par_t1 LSB	par_t1 MSB	par_g2 LSB	par_g2 MSB	par_g1	par_g3	
Fx	variant															

FT260EV1 TO BME680

- µUSB - µUSB, Prime - FT260EV1
- Four wires to connect the FT260EV1 to BME680:

  3v3 - 2-5V GND - GND
  IO0 - SCL IO1 - SDA

- No jumper changes
- No additional PSU required

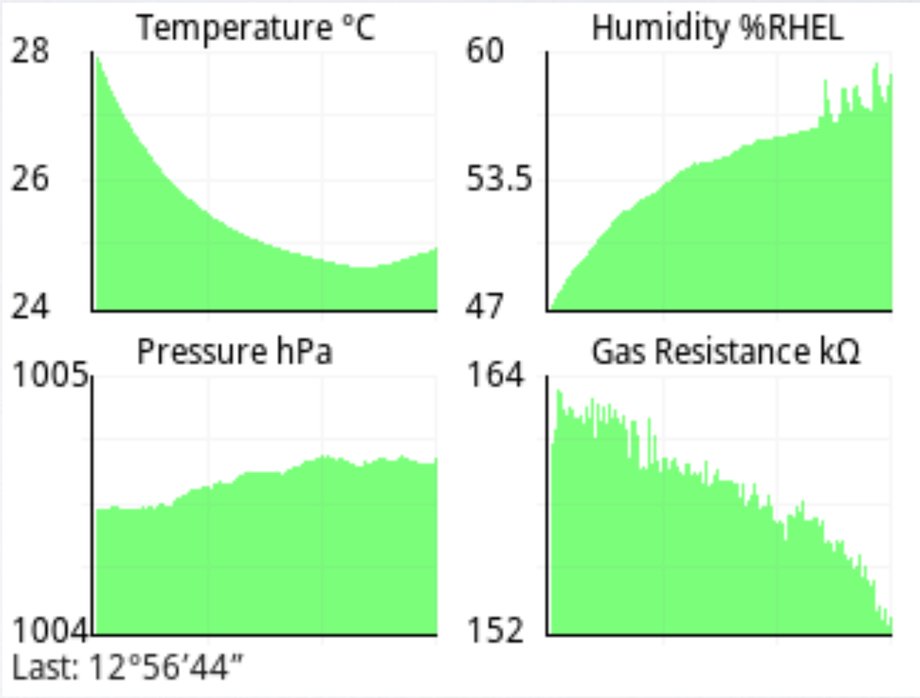


PRIME SOFTWARE

- Built on the FT260 sample code on MoHPC:
  - https://www.hpmuseum.org/forum/thread-17848.html
- Modified based on Bosch C github code:
  - https://github.com/BoschSensortec/BME68x-Sensor-API

PRIME GUI

- bme680_gui() shows autoscaled graphs in real-time
- Press and hold keys: S(low) refresh 10s; F(ast) 1s; D(ebug); Esc to exit and return a results matrix
- bme680_results() takes a matrix of results and shows the graphs
- bme680_term() displays data in the Prime console window (On ➗ and On ✖ toggle the console)



WHAT NEXT?

- Approximately 1100 lines of code - more than

- Time of Flight Sensor - VL53L3CX
  - One measurement - simple? NO!
- Hint: read the API, sample code and manuals before buying the next sensor

ANALOG DEVICES ADXL343

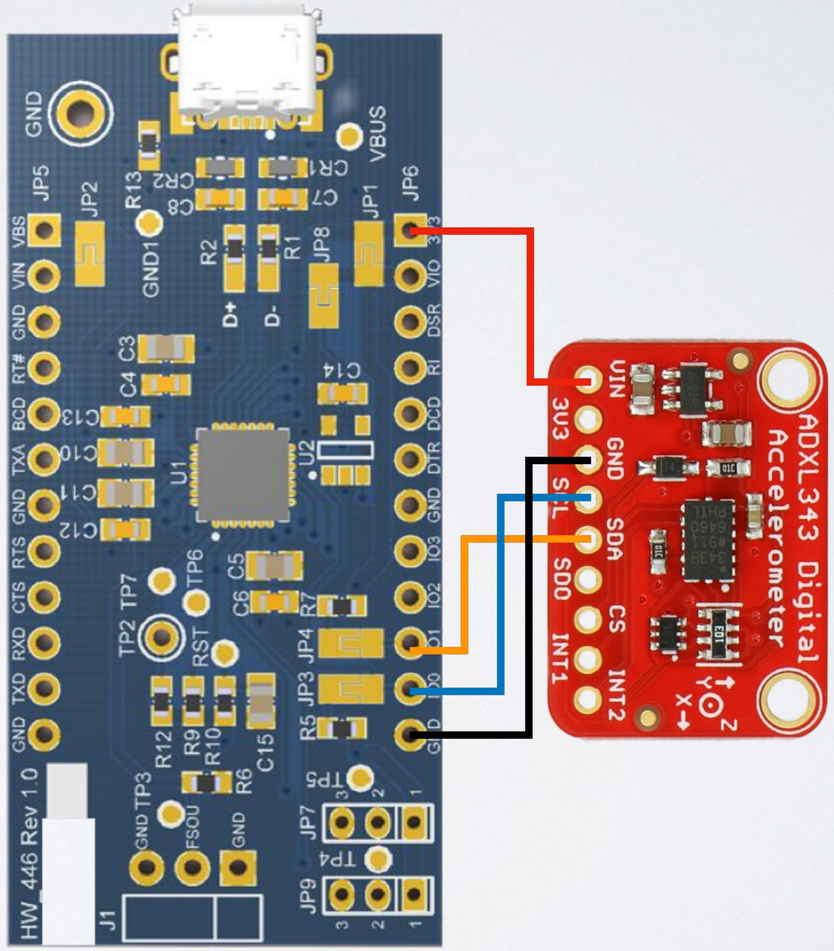
- 3-Axis, ±2 g/±4 g/±8 g/±16 g Digital Micro Electro Mechanical Accelerometer
- Other features: Activity and inactivity sensing detect the presence or lack of motion. Single and double tap detection in any direction. Free-fall sensing.
- Uses include: handsets, gaming and pointing devices, HDD protection
- 3mm x 5mm x 1mm
- Adafruit ADXL343 QT from DigiKey US $5.95, PiHut UK £5.70
- 37-page breakout learning guide and 35-page data sheet
- Simple to configure and use x, y and z data values with ~12 new lines of code

FT260EV1 TO ADXL343

- µUSB - µUSB, Prime - FT260EV1
- Four wires to connect the FT260EV1 to ADXL343:

  3v3 - VIN GND - GND
  IO0 - SCL IO1 - SDA

- No jumper changes
- Daisy-chain with BME680

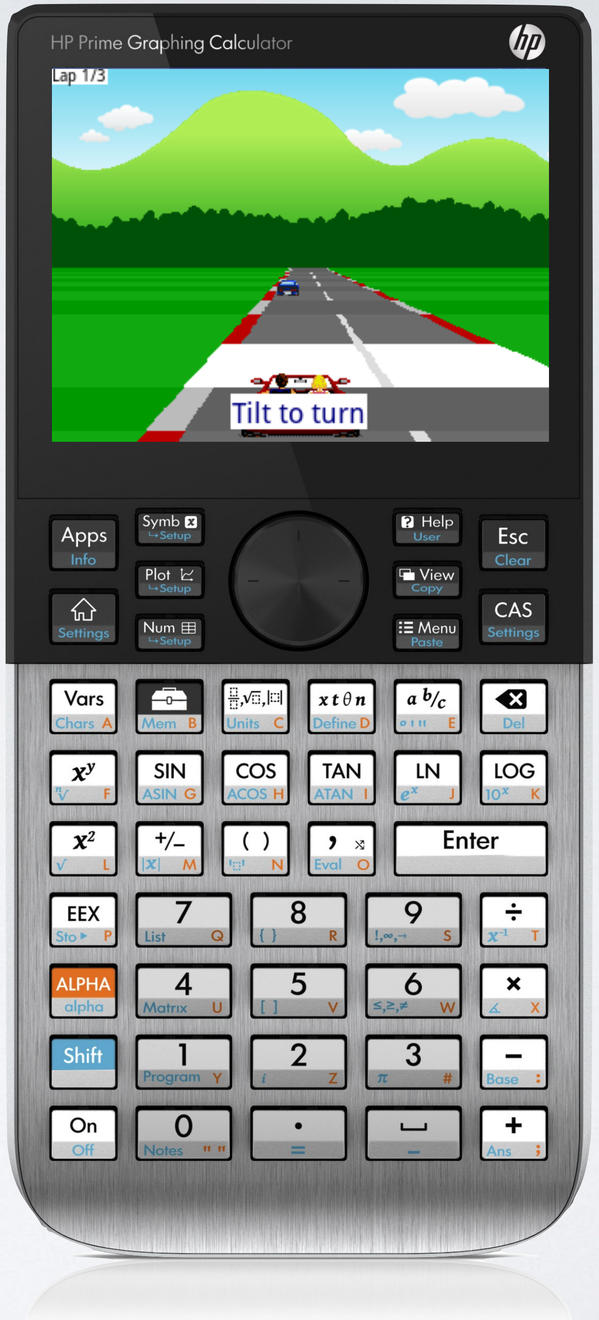


TILT LIBRARY

- Tilt.Init() - checks if a FT260 and accelerometer is connected and sets Tilt.tiltEnabled flag
- Tilt.Read() - returns {x, y, z} each in range -1..1
- Tilt.InvertX(), InvertY(), InvertZ() for convenience
- Easy to add to any program where tilt sensing might be useful

RACER V40

- Increased draw depth due to G2 CPU power
- Tilt to turn with FT260 and ADXL343



SUMMARY

- FT260EV1 + i2c sensors
- Take care with your choice of sensor
- Simple to wire up
- Driven from HP Prime with recent firmware
- No Arduino or Raspberry Pi needed
- Open Source Software