Developers

(/)

ARTIK 0 ARTIK 5/7/10

Sensors with ARTIK IDE

ARTIK modules provide connection capability for analog inputs and various digital input and outputs, as well as I^2C and SPI connections. For the ARTIK 530 and 710 development kits, we offer a special board with a variety of sensors to make testing and development easier. Other board kits do not have matching headers to attach this board.

Here we talk first about the sensor board and some ways to check its function through the command line. Then we show you the easy way, using the ARTIK IDE and its pre-built sensor demo application.

Looking for Python code? You'll find it in our Programmable Pins article (/documentation/developer-guide/gpio/kernel-gpio.html#isup2supc--using-python).

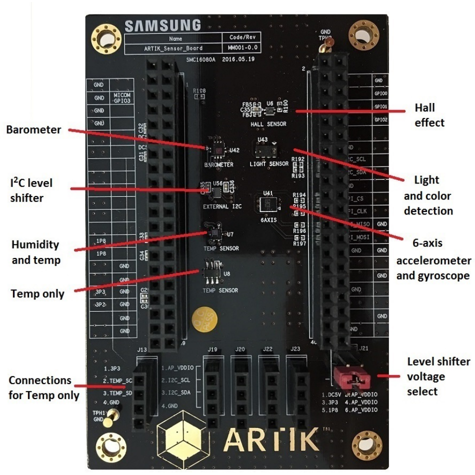
ARTIK Sensor Board

The ARTIK 530 and 710 Interface board provides an option for attaching a Sensor Board, which you can buy through Digi-Key (http://www.digikey.com/product-detail/en/samsung-semiconductor-inc/SIP-ASRNXS004/1683-1009-ND/6236175).

Sensor board – [SIP-ASRNXS004 (https://www.digikey.com/product-detail/en/samsung-semiconductor-inc/SIP-ASRNXS004/1683-1009-ND/6236175)](https://www.digikey.com/product-detail/en/samsung-semiconductor-inc/SIP-ASRNXS004/1683-1009-ND/6236175)
Plugs onto Interface board

Sensors:
- 6-axis motion detection
- Barometric pressure
- Humidity/Temperature
- Temperature
- Hall effect (magnetic)
- Proximity & Color

I^2C Port Expander:
- 4 connectors for external I^2C
- Level shifter
- 5V/3.3V/1.8V selectable



DEVICE FUNCTION	MODEL	ACCESS METHOD	DATASHEET
MEMs barometric pressure sensor	LPS25HB	I^2C bus 1 **0x5D**	ST Micro (http://www.st.com
Humidity/Temperature sensor	HTS221	I^2C bus 1 **0x5F**	ST Micro (http://www.st.com
Temperature sensor	S-5851AAA-M6T1U	I^2C bus 1 **0x48***	SII Semiconductor ic.com/en/tempera
Proximity & Color sensor	CM36655	I^2C bus 1 **0x60** INT on GPIO pin [1]	Capella Microsyste (http://www.capell id=80&mode=16&
Hall Effect sensor	S-5712CCDL1-I4T1U	GPIO pin [2] =0 when magnet near	SII Semiconductor ic.com/en/hall_ics/
6-axis motion detector (3 accelerometer, 3 gyroscope)	LSM6DSM	SPI bus, INT on GPIO pin [0]	ST Micro (http://www.st.com
I^2C Port Expander / level translator	NLSX4373MUTAG	Use **J21** to select voltage	ON Semiconductor (https://www.onse

*The U8 temperature sensor arrives unconnected. Refer [here (/documentation/artik/tutorials/sensor-tutorial.html#temperature-sensor)](/documentation/artik/tutorials/sensor-tutorial.html#temperature-sensor) for details.

Barometric Pressure Sensor

The barometric pressure sensor is wired to respond on I2C bus 1 address **0x5D**. Its register map from the datasheet is shown. Registers 0x20, 0x21, and 0x28-2A are of interest.

Table 17. Registers address map

Name	Type	Register Address Hex	Default Binary	Function and comment
Reserved		00-07 0D - 0E	-	Reserved
REF_P_XL	R/W	08	00000000	Reference pressure registers
REF_P_L	R/W	09	00000000	
REF_P_H	R/W	0A	00000000	
WHO_AM_I	R	0F	10111101	Who am I register
RES_CONF	R/W	10	00001111	Resolution register
Reserved		11-1F	-	Reserved
CTRL_REG1	R/W	20	00000000	Control registers
CTRL_REG2	R/W	21	00000000	
CTRL_REG3	R/W	22	00000000	
CTRL_REG4	R/W	23	00000000	
INTERRUPT_CFG	R/W	24	00000000	Interrupt registers
INT_SOURCE	R	25	00000000, output	
Reserved		26	-	Reserved
STATUS_REG	R	27	00000000, output	Status register
PRESS_OUT_XL	R	28	output	Pressure output registers
PRESS_OUT_L	R	29	output	
PRESS_OUT_H	R	2A	output	
TEMP_OUT_L	R	2B	output	Temperature output registers
TEMP_OUT_H	R	2C	output	
Reserved		2D	-	Reserved
FIFO_CTRL	R/W	2E	00000000	FIFO configure registers
FIFO_STATUS	R	2F	00100000, output	
THS_P_L	R/W	30	00000000	Pressure threshold registers
THS_P_H	R/W	31	00000000	
Reserved		32-38	-	Reserved
RPDS_L	R/W	39	00000000	Pressure offset registers
RPDS_H	R/W	3A	00000000	

Read the current barometric pressure as follows.

1. Identify it as the correct device.
   ```
   i2cget -f -y 1 0x5D 0x0F
   ```
   You should get 'BD'

2. Write CTRL_REG2 to reboot and reset the device.
   ```
   i2cset -f -y 1 0x5D 0x21 0xC4
   ```

3. Write CTRL_REG1 to put the device in active mode and acquire new data every 1s.
   `i2cset -f -y 1 0x5D 0x20 0x90`

4. Read pressure count.
   `i2cget -f -y 1 0x5D 0x2A` high byte
   `i2cget -f -y 1 0x5D 0x29` middle byte
   `i2cget -f -y 1 0x5D 0x28` low byte

   Use 4096 counts per hectopascal (hPa) to convert. For example:

   ```
   3f ba 2e
   ```

   is 4176430 decimal. Divide by 4096 counts/hPa to obtain 1019 hPa, or 30.18 inches of mercury on a U.S. barometric scale.

Humidity / Temperature Sensor

The humidity/temperature sensor is wired to respond on I2C bus 1 address **0x5F**. Its register map from the datasheet is shown. Each device is uniquely calibrated and requires you to interpolate each raw value read using stored values – 14 registers in all!

As a result, we won't be reading back final temperature and humidity values, only the raw measured ones. Refer to the ST Micro datasheet (http://www.st.com/resource/en/datasheet/DM00116291.pdf) for the calibration theory. Search online for the part number to find coding examples in C++ and Python.

Table 15. Register address map

Name	Type	Register address (hex)	Default (hex)
Reserved		00-0E	Do not modify
WHO_AM_I	R	0F	BC
AV_CONF	R/W	10	1B
Reserved		11-1C	Do not modify
CTRL_REG1	R/W	20	0
CTRL_REG2	R/W	21	0
CTRL_REG3	R/W	22	0
Reserved		23-26	Do not modify
STATUS_REG	R	27	0
HUMIDITY_OUT_L	R	28	Output
HUMIDITY_OUT_H	R	29	Output
TEMP_OUT_L	R	2A	Output
TEMP_OUT_H	R	2B	Output
Reserved		2C-2F	Do not modify
CALIB_0..F	R/W	30-3F	Do not modify

Read the device values as follows.

1. Identify it as the correct device.
   `i2cget -f -y 1 0x5F 0x0F`
   You should get 'BC'

2. Write CTRL_REG2 to reboot and reset the device.
   `i2cset -f -y 1 0x5F 0x21 0x80`

3. Write CTRL_REG1 to put the device in active mode and acquire new data every 1s.
   `i2cset -f -y 1 0x5F 0x20 0x81`

4. Read raw humidity value.
   `i2cget -f -y 1 0x5F 0x29` high byte
   `i2cget -f -y 1 0x5F 0x28` low byte

5. Read raw temperature value.
   `i2cget -f -y 1 0x5F 0x2B` high byte
   `i2cget -f -y 1 0x5F 0x2A` low byte

   You can divide by 8 to get the uncalibrated Celsius temperature. For example:

   ```
   00 bc
   ```

   is 188 decimal, divided by 8 is 23.5 C, or 74.3 F

Temperature Sensor

The U8 temperature sensor has no on-board I^2C connection. However, it is configured to respond on I2C address **0x48**. You can wire its SCL and SDA lines from J13 over to J19 to I^2C bus 1 on the translated side of the I2C expander – but make sure you first put a jumper on J21 to select an appropriate level shifter voltage (typically 3.3V).

Read the device values as follows.

1. Enable the device by writing 0 to its Configuration register at index 01.
   `i2cset -f -y 1 0x48 0x1 0`

2. Read the raw temperature value in hex at index 00 as a word (two bytes).
   `i2cget -f -y 1 0x48 0x0 w`
   (returns bits[3:0]+0000 – bits[11:4])

3. Swap the hex bytes to get MSB – LSB order.

4. Right-shift the 16-bit value to return the final 12-bit Celsius temperature.

For example, a returned raw value of:

```
0x6019
```

refers to the low byte of '0x60' and high byte of '0x19'.

– Swapped, it becomes 0x1960

– Right-shifted, it becomes 0x196

– Converted to decimal, it becomes 406

– Multiplied by .0625 count / degree C, you arrive at the final value of 25.4 C.

Hall Effect Sensor

The Hall effect sensor has no programmability. It simply triggers its output connected to GPIO pin [2], which is `sysfs GPIO 130`, when a magnet is placed nearby. To test:

1. Request control of the desired GPIO.
   ```
   echo 130 > /sys/class/gpio/export
   ```
2. Configure the GPIO to be an input.
   ```
   echo in > /sys/class/gpio/gpio130/direction
   ```
3. Read its value.
   ```
   cat /sys/class/gpio/gpio130/value
   ```

Try this last step repeatedly, placing a magnet near the sensor and then drawing it away, to see the difference in the return value.

- `1`: no magnet
- `0`: magnet detected

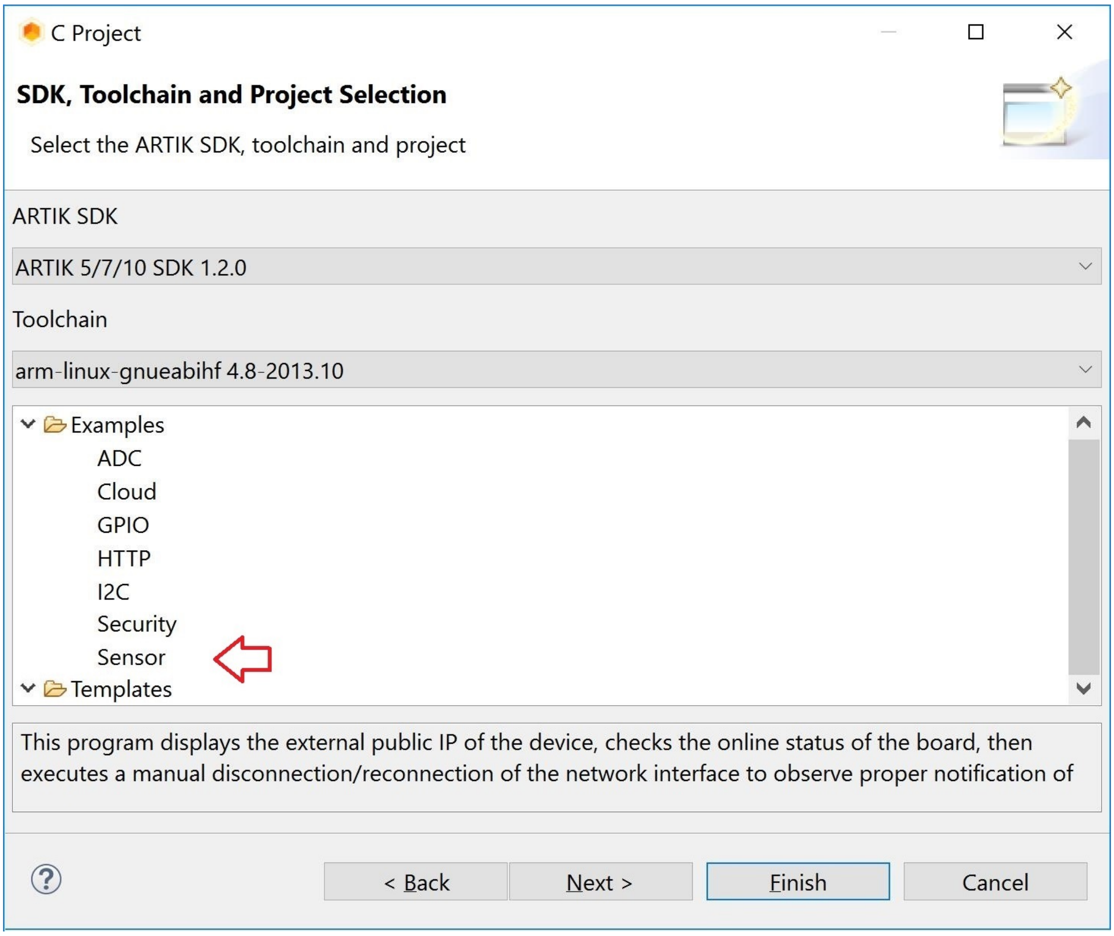
Proximity and Color Sensor

The CM36655 device is related to the CM36653 device, for which sample code can be found [here (https://github.com/Fuzion24/SM-G900V_NA_KK_Opensource-S5-Kernel-/blob/master/drivers/sensors/cm36653.c)](https://github.com/Fuzion24/SM-G900V_NA_KK_Opensource-S5-Kernel-/blob/master/drivers/sensors/cm36653.c).

ARTIK IDE Sensor Demo

You're ready to try out an ARTIK IDE development project!

Go to [ARTIK IDE – Linux Projects (/documentation/developer-guide/artik-ide/linux-apps.html)](/documentation/developer-guide/artik-ide/linux-apps.html) and follow the instructions. When you get to the screen that lets you choose the application you want to develop, choose **Sensor**.



You'll find that code is provided for all sensors except for the Hall Effect sensor, which is left for you to implement. Have fun!

Last updated on: Sep 26 2017, 5:37 PM

Developers (https://www.artik.io/get-started)

Events calendar (https://www.artik.io/events)

Media (https://www.artik.io/media)

Samsung.com (http://www.samsung.com)