thinxtra
Empowering Internet of Things

sigfox operator

Thinxtra Xkit Development Guide for Arduino

April 2017

www.thinxtra.com/xkit

INSTRUCTIONS & PREREQUISITES

This is a documentation to help you developing specific applications with the Thinxtra Xkit shield and an Arduino Uno R3 board . The document describes how to setup and use the Arduino Interactive Development Environment (IDE), then how to you the firmware libraries to program the Arduino board.

To be able to complete the flow, the following pre-requisites are required:

- Register your Thinxtra Xkit on Sigfox backend (refers to “First Steps Xkit”)

INSTALLING ARDUINO IDE

thinxtra Empowering Internet of Things | ARDUINO

To develop application with Xkit shield via Arduino board, it is recommended to use Arduino IDE for its simplicity.

- Download and install the standard Arduino environment for your operating system (Windows, Mac OS X, Linux or Portable IDE)
- The Arduino IDE download page is https://www.arduino.cc/en/Main/Software.
- After successful installation, start Arduino IDE
- Check in the menu: *Tools* | *Board*, that the type of board selected is *Arduino / Genuino Uno*
- Check in the menu: *Tools* | *Programmer*, that the type of programmer selected is *AVRISP mkII*

INSTALLING XKIT ARDUINO LIBRARIES

thinxtra Empowering Internet of Things | GitHub

- Download the Xkit Arduino firmware repository as a zip file from https://github.com/Thinxtra/Xkit-Sample
- Unarchive. You only require the libraries directory (including the folders Isigfox, SimpleTimer and Tsensors).
- Copy the content of the libraries directory into:
  - *C:\Users\<yourusername>\Documents\Arduino\libraries* on Windows
  - */Users/<yourusername>/Documents/Arduino/libraries* on OS X
- Restart the Arduino IDE

ADDING XKIT ARDUINO LIBRARIES TO SKETCH

- The downloaded Xkit Arduino libraries should now appear under *Contributed Libraries* when you click on *Sketch* | *Include Library*.
- To add libraries to your sketch, you can:
  - Click on the Xkit Arduino libraries added under *Sketch* | *Include* | *Contributed Libraries*
  - Add the following lines of code at the top of your sketch:

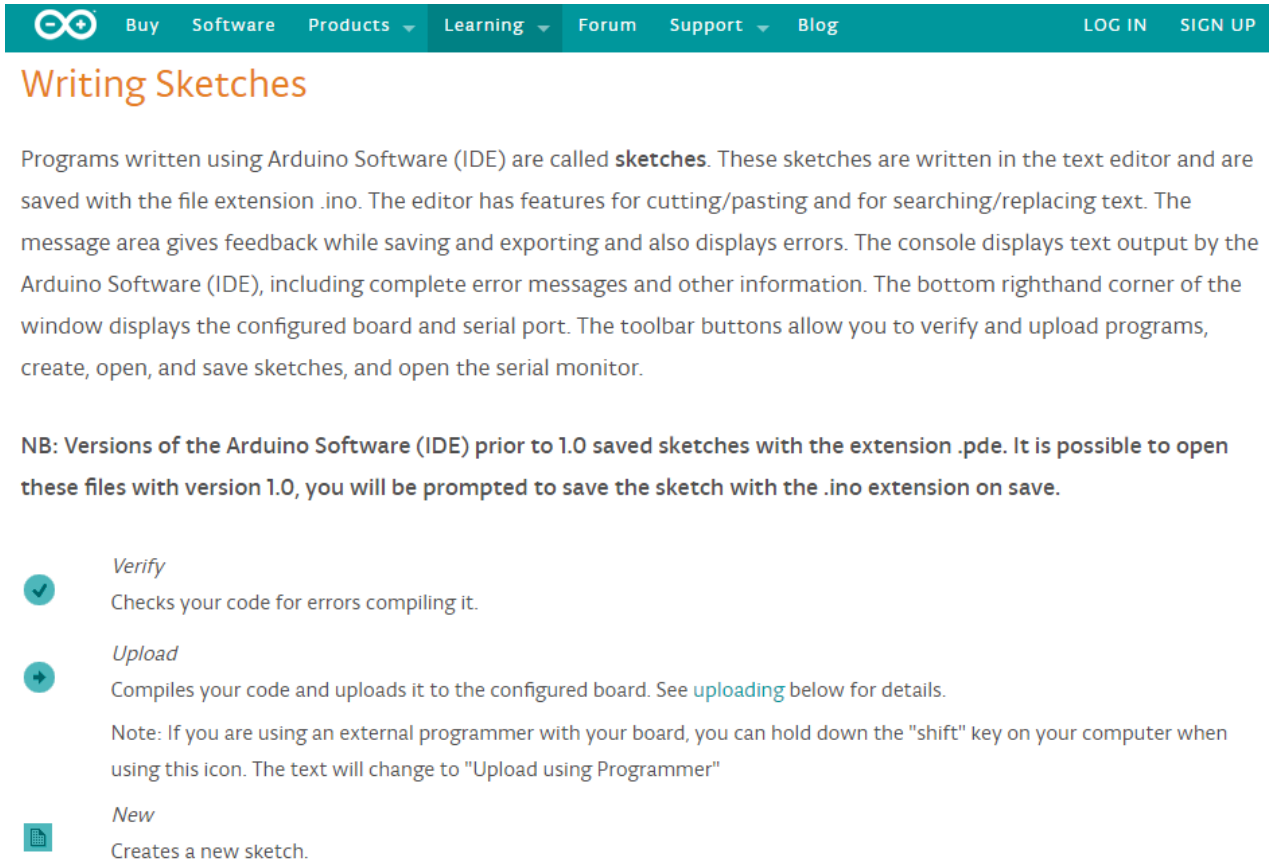
    #include <WISOL.h>

    #include <Tsensors.h>

HOW TO USE ARDUINO IDE

- You can find information on Arduino IDE, on Arduino website: https://www.arduino.cc/en/Guide/Environment

- **Writing Sketches** section is indeed essential and you should go through it before going further in this document. You can either:
  - Start “from scratch” and write your own sketch or,
  - Use the DemoApp, provided by Thinxtra (you should have already downloaded the source code with the libraries from the Thinxtra github account) and modify it to fit your application



HOW TO USE ARDUINO IDE

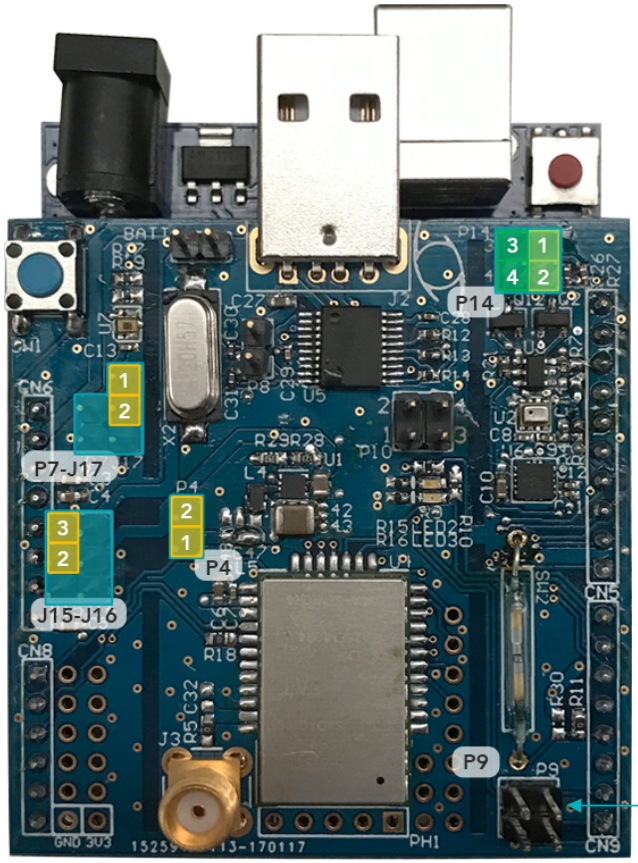
The following picture shows an example window from Arduino IDE

- The right-arrow icon on the upper left corner as circled in the picture is to upload the current sketch to the Arduino board. If the sketch is uploaded successfully, the message “Done uploading” is shown as circled in the picture.

- The magnifier icon on the upper right corner as circled in the picture is to open the Serial Monitor which is a simple way to observe messages and interact with Arduino board via UART protocol.

- Arduino IDE occasionally cannot detect the correct COM port connecting to Arduino board, which can lead to an error when uploading sketch**.** In such cases, go to Tools > Port: “ ” > and choose the correct COM port.

IMPORTANT NOTICE WHEN UPLOADING A SKETCH TO ARDUINO BOARD



- To upload a sketch to the Arduino board (with the Xkit shield plugged into it), you must remove the P9 jumpers on the Xkit shield as described in the **left** picture.
- Once the sketch is successfully uploaded, reconnect the P9 jumpers.

Note: Arduino IDE Serial Monitor output will not be received by the Arduino board (disabling sketch upload) if the above description is not respected.
Alternative solution: simply disconnect the Xkit shield from Arduino board when uploaded a sketch.

ISIGFOX LIBRARY - OVERVIEW

This library provides functions to initialise and configure Sigfox communication. In addition, it also implements functions to send Sigfox messages.

NOTES

- SIGFOX global network is divided in 4 regions called: **RCZ1, RCZ2, RCZ3 and RCZ4.**
- Thinxtra Xkit supported region is indicated on its packaging or on its radio module
- RCZ4 includes Australia, New Zealand, South East Asia countries (except South Korea and Japan) and LATAM countries (except Brazil)
- More information on the 4 different regions on Thinxtra website: http://www.thinxtra.com/devicemakers/
- You should add the Isigfox library as a *Contributed Libraries* (refers *Adding Xkit Arduino libraries to sketch* section in this document), you should include WISOL header (itself including Isigfox header) in your Arduino sketch by adding: *#include <WISOL.h>*

ISIGFOX LIBRARY – HOW TO USE

Function Prototype	int initSigfox()
Description	Set the zone for Sigfox communication. This function should be called during the setup phase of the device – to configure it correctly.
Parameters	None
Return Value	0 if succeed and -1, otherwise
Notes	RCZ3 is not supported in V.1.0 of the Isigfox library

Function Prototype	int getZone()
Description	Get the zone of the Sigfox module
Parameters	None
Return Value	{1, 2, 4}, respectively for RCZ1, RCZ2 or RCZ4. Other values are invalid
Notes	RCZ3 is not supported in V.1.0 of the Isigfox library

ISIGFOX LIBRARY – HOW TO USE

Function Prototype	int setZone()
Description	Set the zone of the Sigfox module
Parameters	None
Return Value	-1
Notes	Function is not implemented in V.1.0 of Isigfox library

Function Prototype	int testComms()
Description	Test UART communication between Arduino board and Sigfox module on the Thinxtra Xkit shield
Parameters	None
Return Value	0 if succeed and -1, otherwise

ISIGFOX LIBRARY – HOW TO USE

Function Prototype	int sendPayload(uint8_t *outData, const uint8_t len, int downlink, recvMsg *receivedMsg) int sendPayload(uint8_t *outData, const uint8_t len, int downlink)
Description	Send a Sigfox frame
Parameters	*char *outData:* pointer on first byte of the payload to be sent over Sigfox network *int len:* [0..12], length of the payload *int downlink:* 0, no downlink message required or 1, downlink message required. Other valid are invalid. *recvMsg *receivedMsg*: pointer on recMsg structure. if NULL or not used, the function will not block until Sigfox module's answer reception.
Return Value	0 if succeed and -1, otherwise
Notes	When sending a frame, the Sigfox module acknowledge the reception by sending an answer – it can take up to 45 seconds to retrieve the message in case downlink message is required. The structure of the received message is: typedef struct _recvMsg{ int len; /*!< the length of the array of received bytes */ char* inData; /*!< a pointer to an array of received bytes */ } recvMsg;

ISIGFOX LIBRARY - TROUBLESHOOT

- You can send a Sigfox message without using Isigfox library functions. To do so, the following strings are needed to be sent through the UART bus according to your region as follows:

  For RCZ1

  - *AT302=15*
  - *AT$SF=<yourpayload>,* where *<yourpayload>* is the payload, expressed in hexadecimal format.

  For RCZ2

  - *AT$SF=<yourpayload>,* where *<yourpayload>* is the payload, expressed in hexadecimal format.

  For RCZ4

  - *AT$RC*
  - *AT$SF=<yourpayload>,* where *<yourpayload>* is the payload, expressed in hexadecimal format.

- You can find more information on UART bus, on Arduino website: https://www.arduino.cc/en/reference/serial
- The minimum payload size is 0 byte. The maximum payload size is 12 bytes.
- The minimum period between two successive Sigfox messages should be equal or superior to 20 seconds

TSENSORS LIBRARY – OVERVIEW

This library provides functions to initialise sensors and retrieve measurement. Thinxtra Xkit offers access to:

- Digital temperature and pressure sensor (Bosh BMP280),
- Accelerometer 3-axis (NXP MMA8415Q)
- Ambient Light Sensor (Panasonic PNJ4K01F)
- Reed Switch

NOTES:

- You should add the Tsensors library as a *Contributed Libraries* (refers *Adding Xkit Arduino libraries to sketch* section in this document), you should include Tsensor header in your Arduino sketch by adding: *#include <Tsensor.h>*
- Arduino float value is 32-bits value. You can find more information on Arduino website: https://www.arduino.cc/en/reference/float

TSENSORS LIBRARY – HOW TO USE

Function Prototype	int initSensors()
Description	Run initialisation process for sensors on Thinxtra Xkit
Parameters	None
Return Value	0 if succeed and -1, otherwise

Function Prototype	float getPressure ()
Description	Get the current ambient temperature
Return Value	Float value of the current ambient pressure (Pa)

Function Prototype	float getTemp ()
Description	Get the current ambient temperature
Parameters	None
Return Value	Float value of the current ambient temperature (Celsius)

TSENSORS LIBRARY – HOW TO USE

Function Prototype	float getPhoto ()
Description	Get the current ambient light
Return Value	a float value of the current output voltage (V)

Function Prototype	int getAccXYZ (acceleration_xyz *acc)
Description	Get the current acceleration on all three axes
Parameters	acceleration_xyz acc: pointer on acceleration_xyz structure. If NULL, function will return an error.
Return Value	0 if succeed and -1, otherwise
Notes	The structure of the output acceleration in 3D typedef struct { float x_g; /*!< The acceleration on the X axis (expressed in g) */ float y_g; /*!< The acceleration on the Y axis (expressed in g) */ float z_g; /*!< The acceleration on the Z axis (expressed in g) */ float a_g; /*!< The 3D magnitude of the acceleration */ } acceleration_xyz;

TSENSORS LIBRARY – HOW TO USE

Function Prototype	float getAccX ()
Description	Get the current acceleration of the X axis
Return Value	a float value of the current acceleration (g) of the X axis

Function Prototype	float getAccY ()
Description	Get the current acceleration of the Y axis
Return Value	a float value of the current acceleration (g) of the Y axis

Function Prototype	float getAccZ ()
Description	Get the current acceleration of the Z axis
Return Value	a float value of the current acceleration (g) of the Z axis

TSENSORS LIBRARY – HOW TO USE

Function Prototype	void setButton(void (*service(void)))
Description	Set a callback which is triggered on a button press
Parameters	*service(void): a function pointer which points to the desired callback
Return Value	None

Function Prototype	void setReed(void (*InterruptService(void)))
Description	Set a callback which is triggered when a magnet is close to the reed switch
Parameters	*InterruptService(void): a function pointer which points to the desired callback
Return Value	None


thinxtra
Empowering Internet of Things
sigfox
operator
www.thinxtra.com/xkit