Near Space Arduino Course

Marc Cortés i Fargas

ESEIAAT, Technical University of Catalonia – **BarcelonaTech**

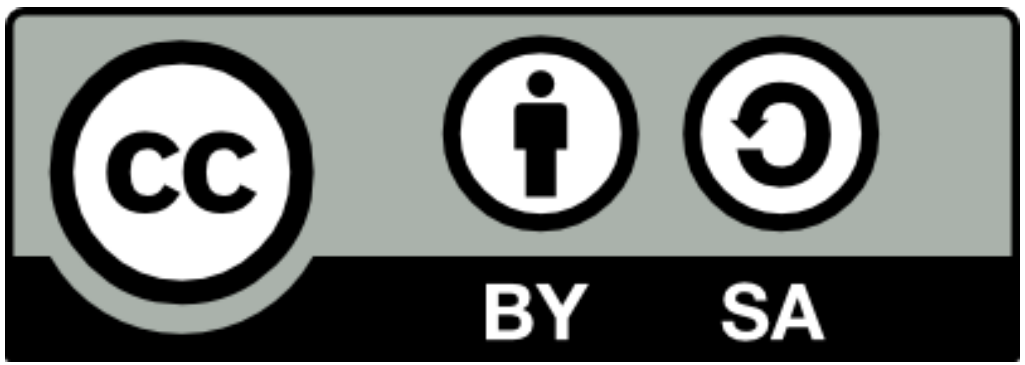
marc.cortes.fargas@upcprogram.space

May 9, 2017

Navigation

You can use sections at the top of the pages to navigate though this document. Also, everything in **bolt text** is a link to an external page.

License



Images of this course

All schemes of this course are drawn by using **Fritzing**.
Also, all the third-images and videos attached are used in a non-commercial way.
The images of section "Protocol communications" are extracted from **Spark Fun**.

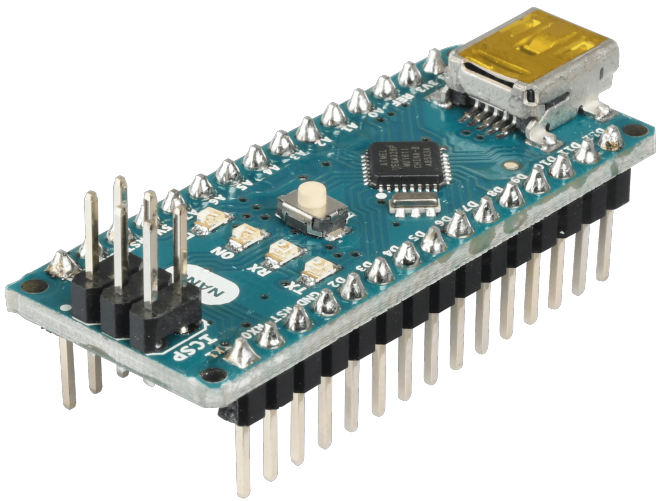
Overview

Contents of the course

Contents

1. ~~Digital signals (IN and OUT) – Pulldowns, Arduino basics, Analog signals – A/D, PWM. *22/03/2017*~~
2. ~~Barometer/Termometer (I2C comms), ADC, Servomotors, Battery dimensioning. *26/04/2017*~~
3. *Bluetooth* issues (Serial comms.), Relay, Switch on a lamp using a mobile phone ~~(only on *Android*)~~ I2C comms. BMP180 *10/05/2017*
4. GPS main sentences. Filtering and main parameters. Communication through **Iridium** satellites. *24/05/2017*

Figure: Arduino nano.



Today's contents

1. Libraries in Arduino
2. **GitHub**, your new friend
3. Adding libraries to Arduino's IDE
4. Communication protocols
5. BMP 180
6. Exercise 1 - Fix library
7. Relay
8. How fast can we switch a relay? - Exercise 2
9. Relay vs MOSFET
10. Serial communications
11. Bluetooth
12. In-class exercise: switch on a lamp by using your own mobile phone.

Libraries

Libraries are external scripts that provide more functions to your IDE. They can be written in other programming languages such as C/C++.
Arduino's libraries usually provide the user basic functions and could implement protocols of communication such as I2C or SPI.

Where can we find libraries?

The greatest website to find Arduino's libraries, among libraries and scripts for other codes is website **GitHub**.
UPC Space Program is on GitHub! Click **here** to see different codes used in aerospace projects.

Add libraries into Arduino's IDE

First, go to **Sparkfun's BMP180 library** and automatically download a ZIP compressed file. This is the library for the barometer-thermometer "BMP180".
Now, you can add a library by following any of these methods:

Method 1

Just go to the Arduino's IDE and: *Sketch/Include Library/Add .ZIP library* and select ZIP file.

Sometimes this method leads to an error. If that occurs...

Method 2

Uncompress the file and copy the folder to your libraries folder. In Windows, go to Documents/Arduino/Libraries.
In Unix based systems, go to home folder and select "sketchbook" folder. Copy there the folder.
Besides that, you have to make sure to not put any "-" in the folder's name. Remember to rename the library's folder if you use this method!

BMP180 - Barometer termometer

BMP180 is a common sensor manufactured by Bosch widely used in industry. If we go to its **datasheet**, we see different parameters:

BMP180 parameters

1. Power supply
2. Sensitivity
3. Applications
4. *Part 5: Protocol of communication !*: I2C comms.

But... what does "Protocol of communication" means?

Communication protocol

A communication protocol is a method to "speak" with other devices. In arduino we will use basically three communication protocols:

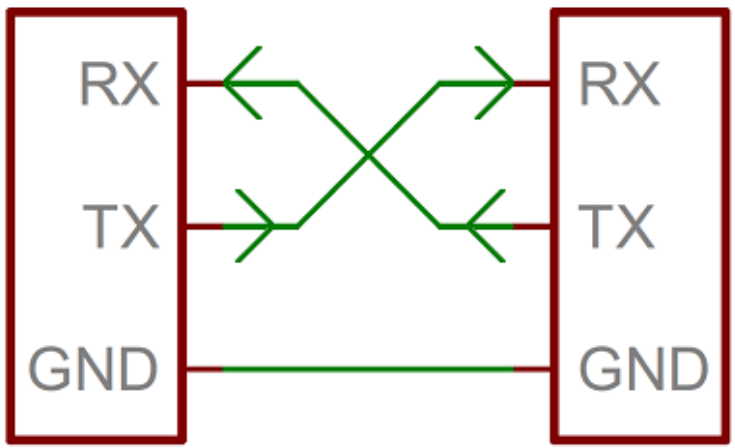
Serial communication

Communication asynchronous widely used in Arduino and other devices such as GPS, Bluetooth module or Iridium Module. This is an asynchronous communication, and that mean all the devices need to have same velocity to speak, aka baud rate.
To start communication from arduino to your computer, you have to start serial communication by using the command *Serial.beginX*, where X is the baud rate.
Baud rate should be the same in your script and in your Serial monitor!

Serial communication

Wiring for serial communication is:
Tx from device 1 to Rx to device 2. Tx means "transmit" and Rx means "receive."
Rx from device 1 to Tx to device 2.
GND from both devices connected.
And, of course, power supply for both devices.

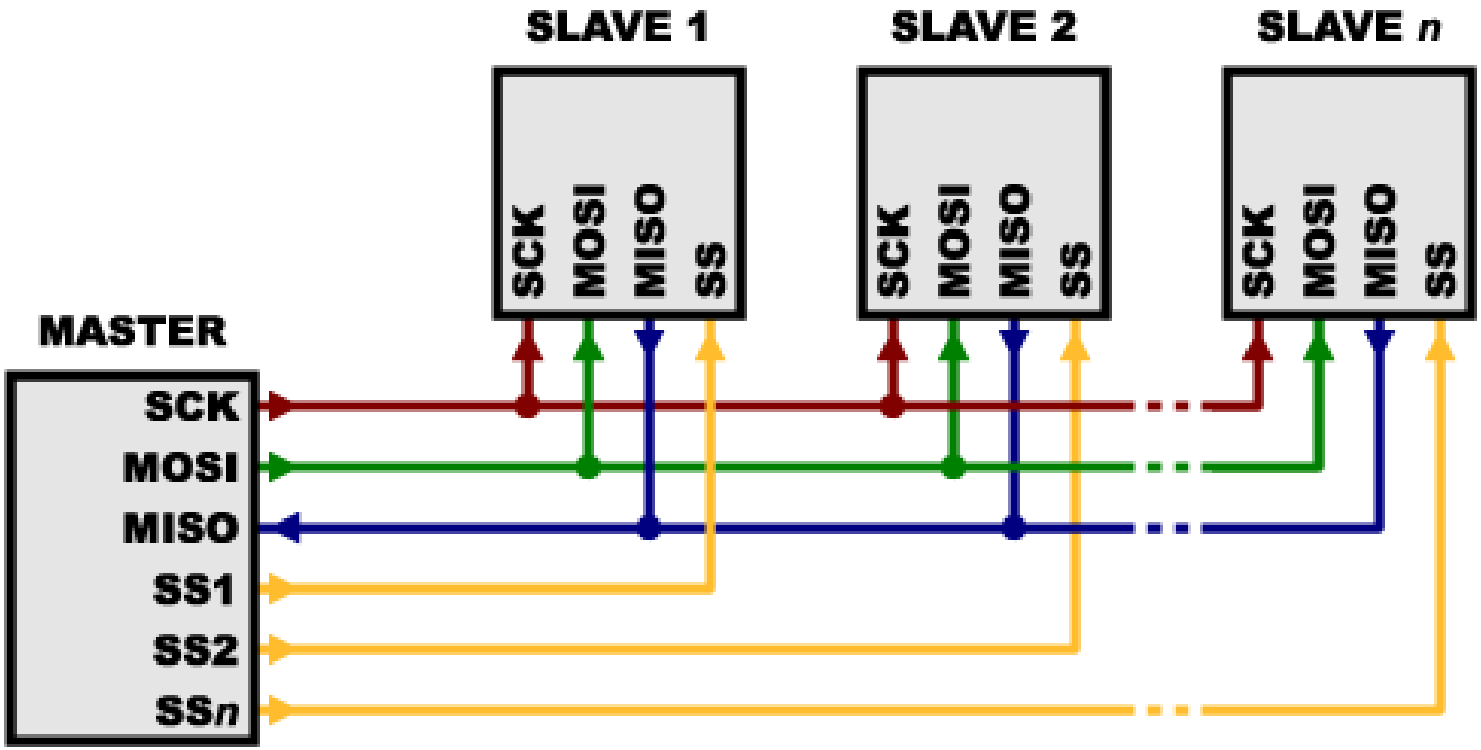
Figure: Serial communication



SPI

SPI is a synchronous protocol of communications used in devices such as SD loggers.
In SPI a device is declared as a master and others as a slaves.
4 pins are used:
SCK: a clock to get the synchronous protocol.
MOSI:Master Out Slave In
MOSI: Master Out Slave In.
CS: Chip Select. That pin will help us to identify each salve.
GND: commomn grounds.

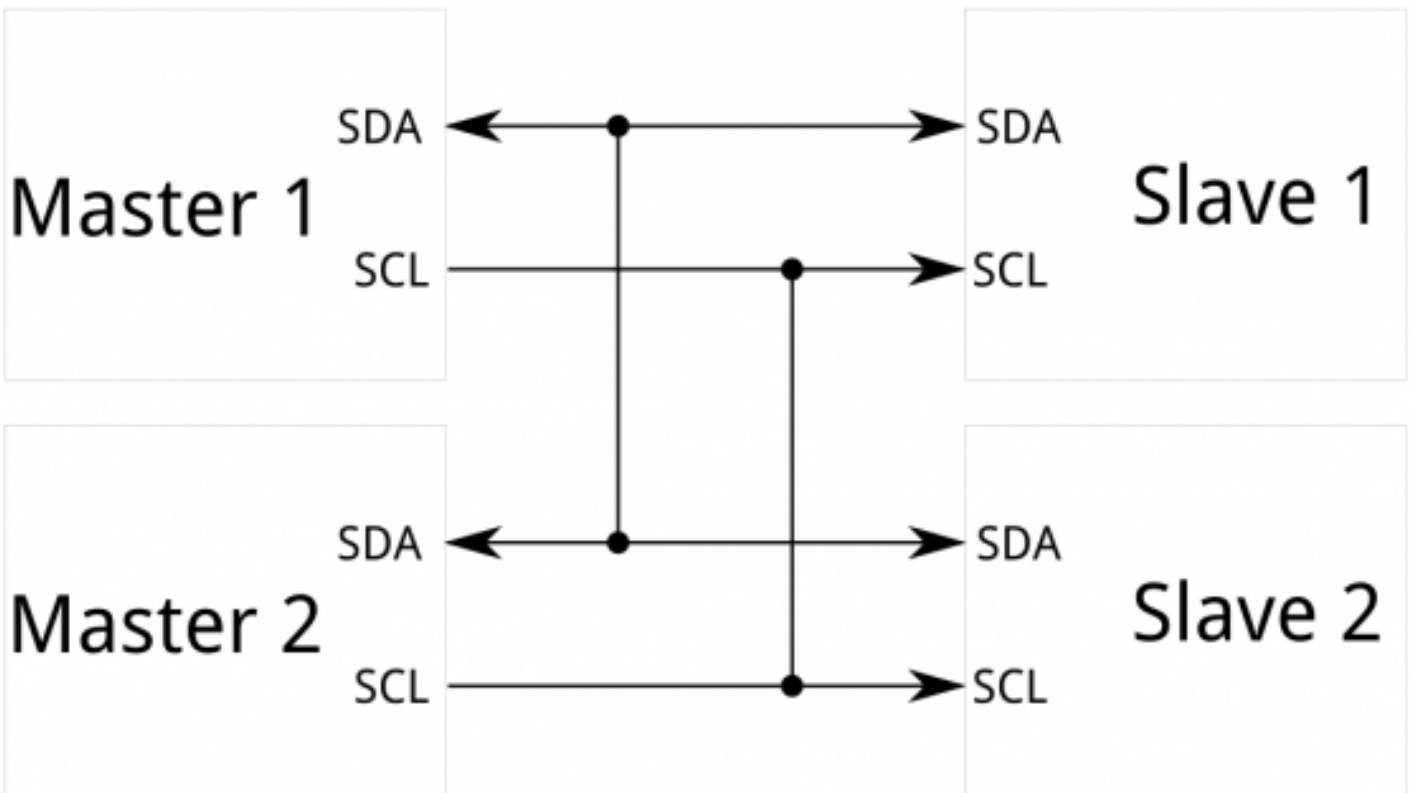
Figure: SPI configuration



I2C

I2C is a synchronous protocol of communications that aims to be the best from serial and SPI.
Serial protocol has the issue that is asynchronous and that you will need an Rx and Tx line for each device that you want to speak with.
SPI communication needs 4 lines for each device and only supports one master device.
I2C only needs 3 lines:
SCL: Serial clock, to get the synchronous communication.
SDA:Serial Data, to send data from/to each device.
GND: as always all GNDs must be connected.

Figure: I2C configuration



I2C issues

I2C speaks with a each device because each I2C device have an own name or address.
However, if you want to communicate with two devices with the same address, that will be impossible in the standard I2C communication.
That means that by default you cannot use two BMP180 with only one Arduino!
Library that handles I2C communication in Arduino is called "wire".

Connection of BMP180

Connect BMP180 following the wiring on the board.
To know which pins are SCL and SDA of your Arduino, go to pin layout in the slides of Lecture 1.
And please, connect VCC to 3V3, *not 5V!*

BMP180 - Example

1. Load *SFE BMP180 EXAMPLE* from examples folder.
2. Activate Serial Monitor.
3. What happens?

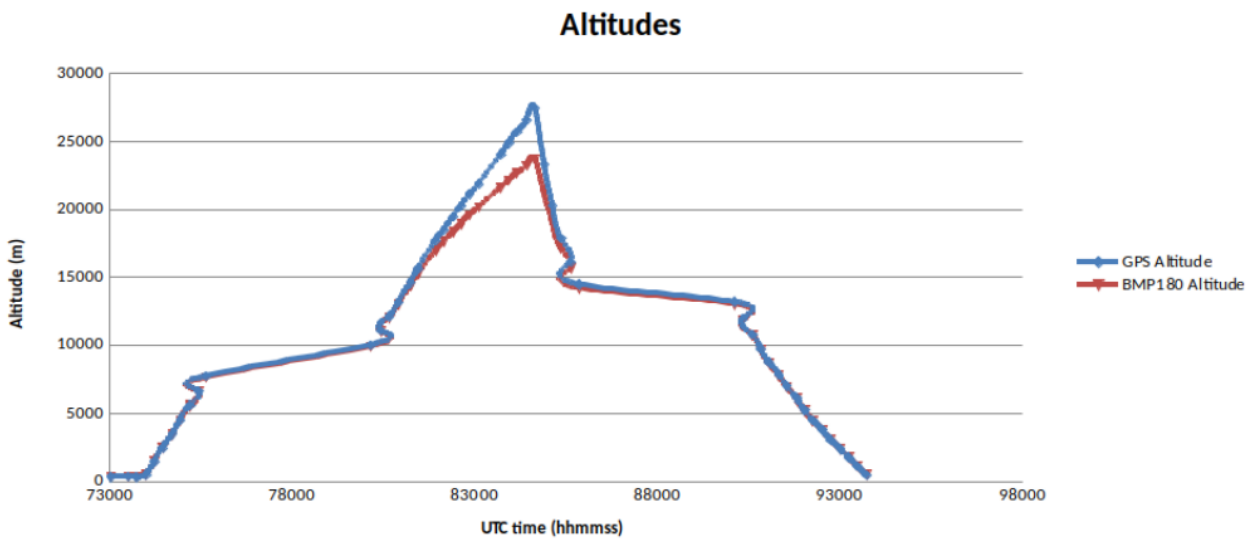
Getting barometric altitude from a reference pressure

Homework: develop a code that returns an altitude by a given pressure.

BMP180 Accuracy in Near Space

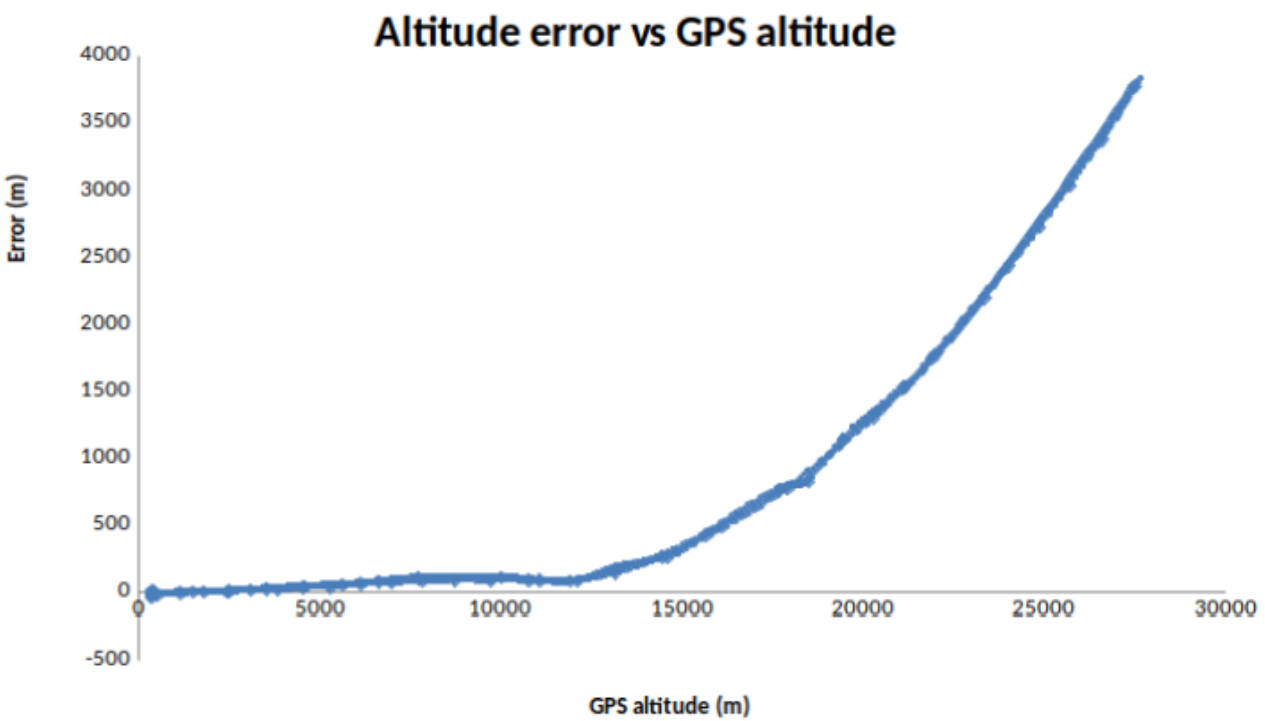
How wrong can be a 2 EUR barometer?
Comparison between an **(uBLOX-M8Q GPS)** (about 30 EUR) and a BMP180.

Figure: Deviation in altitude in a HAB mission in April'17.



Deviation. Detail.

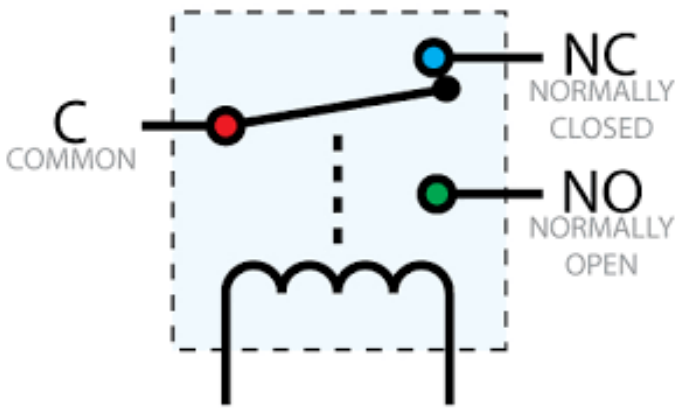
Figure: Deviation in altitude (2)



Rely on the relay

A relay is an electromechanic device used to turn on a secondary circuit where high power is needed from a signal-based circuit. A coil is activated and the magnetic field created moves a metal stick that can switch on/off a circuit.
Tl, dr., It is like a switch but activated though an electrical signal.

Figure: Relay scheme



How fast can we switch a relay?

Let's test our relay!

1. Connext relay using jumper wires. GND to GND, VCC to VCC and IN to a digital pin of your arduino.
2. Load "blink" code and play with the delays to get the fastest switching velocity available. *At which frequency do we achieve that?*

Relays in aerospace industry

- Relays are vastly used in airplanes. They are used, i.e., to indicate flap positions. However, if they fail they have **terrible consequencies.**
- Relays are not used in space applications.
- Relay's lifetime is closely related to the time we activate it. In space, we cannot change a relay!

RELAY

There is an electronic device called MOSFET used to switch on/off a secondary circuit. When we should use a MOSFET or a relay?

Relay

- In low-budget situations.
- In heavy duty applications.
- When we want completely independence of activator circuit and activated circuit.
- When need to easily replace the component
- It is used in quotidian applications.

MOSFET

MOSFET

- In Space environments, where you cannot replace the component.
- In Near Space environments. Air's conductivity changes as a function of the altitude, and an incremental of the air's conductivity can activate the relay spontaneously: **REF1**.
- In small devices.
- When we need a signal to be activated/deactivated in a very short time! (i.e., PWM).

What is Bluetooth?

Bluetooth is a standard of communications widely used in mobile phones.
There are several modules that can be bought. In this course we will use a bluetooth module BLE (Bluetooth Low Energy) and this module will communicate to your arduino through Serial communication.

Figure: Why one should use Bluetooth?

Mobile app connection.

Now, to use your bluetooth module to start a communication with your Arduino:

1. Connect GND to GND, 5V to VCC and 5V also to EN (enable).
   Do not connect RX and TX
2. Pair your device with the module. Module name is "MLT-BT05" and default password is "123456".
3. Download a bluetooth terminal app. For Android, use **BLE Scanner**. MLT-BT05 module works with iOS devices: download a BLE bluetooth terminal app from AppStore or go **here**.
4. Download example code from **GitHub**.

Script discussion

- Do you think is necessary to use *Software Serial*?
- How does the script save and do the data treatment?

Try to change code to familiarize with Serial communication. I.e., send via bluetooth "What's your name?" And make your Arduino send you your own name. Make sure to connect Rx and Tx pins to the defined in *SoftwareSerial* !

Exercise Bluetooth

Create a script to turn on and off a LED.
Once completed, change the led by a relay.
Ask your teacher for a demonstration with a real lamp!

References

Jeremy Blum

Exploring Arduino

Ed. Wiley

Robert Boylestad et al

Electronic devices and circuit theory

Ed. PHI, 1982

Margolis, Michael

Arduino cookbook

Sebastopol, CA : O'Reilly, cop. 2012, 2nd ed.

You can find this book online **here**, only for UPC community.