

A practical component-oriented approach to IoT design and implementation

for students in Computer Science

smartcomputerlab.org

P.Bakowski

Table of Contents

1. Introduction	
1.1 Specification blocks – symbols for IoT systems.....	3
1.2 A few simple examples of IoT architectures.....	4
1.3 IoT systems – hardware and software.....	5
1.3.1 The Arduino Hardware.....	5
1.3.2 Arduino IDE.....	5
1.3.2.1 Arduino IDE installation.....	7
1.3.2.2 Arduino IDE libraries and sketches.....	7
1.3.3 C, C++ and Arduino (code) programming.....	9
1.3.3.1 I/O functions.....	9
1.3.3.2 Time functions.....	10
1.3.3.3 Mathematical functions.....	11
1.3.3.4 Interrupts.....	12
2. Connection and communication technologies.....	13
Introduction.....	13
2.1 Essential connection links and buses.....	13
2.1.1 Asynchronous serial communication.....	13
2.1.1.1 UART Receiver.....	13
2.1.1.2 UART Transmitter.....	14
2.1.2 I2C bus.....	16
2.1.2.1 I2C bus structure.....	16
2.1.2.2 I2C bus operational mode.....	16
2.1.2.3 I2C bus with Arduinos and ESPs.....	17
2.1.2.4 Arduino I2C programming (master and slave).....	17
2.1.3 SPI bus.....	21
2.1.3.1 SPI bus connection.....	21
2.1.3.2 SPI bus operation.....	22
2.1.3.3 SPI bus connections and operation on Arduino/ESP boards.....	22
2.1.3.3 Interrupts.....	24
2.2 Essential wired and wireless communication channels.....	25
2.2.1 Ethernet (IEEE 802.3).....	25
2.2.1.1 Ethernet frame structure.....	25
2.2.1.2 Arduino Ethernet (network) programming.....	26
2.2.2 WiFi (IEEE 802.11.x).....	33
2.2.2.1 WiFi 802.11.g efficiency and channels.....	33
2.2.2.2 WiFi security.....	33
2.2.2.3 SSID and RSSI.....	34
2.2.2.4 Arduino and ESP8266 WiFi.....	34
2.2.3 RF433 – simple radio channels on 433MHz band.....	42
2.2.3.1 RF433 modules.....	42
2.2.3.2 An example of Arduino code for RF433 transmitter and receiver.....	42

2.2.3.3	An example of Arduino code for RF433 transmitter for switch control.....	44
2.2.4	HC-12–Low power radio channels on 433MHz band.....	45
2.2.4.1	An example of Arduino code for HC-12 (443MHz).....	45
2.2.5	NRF24 – simple radio channels on 2.4GHz band.....	46
2.2.5.1	nRF24 modulation.....	46
2.2.5.2	nRF24 modules.....	46
2.2.5.3	An example of Arduino code with nRF24L01.h and RF24.h libraries.....	47
2.2.6	Bluetooth link on 2.4 GHz band.....	48
2.2.6.1	Bluetooth modules.....	48
2.2.6.2	An example of Arduino code with SoftwareSerial.h library.....	48
2.2.7	Long Range communication with LoRa (433 MHz, 968MHz).....	49
2.2.7.1	Introduction to LoRa technology.....	49
2.2.7.2	Packet format.....	51
2.2.7.3	Connections and pins.....	51
2.2.7.4	Interrupts.....	51
2.2.7.5	Memory.....	51
2.2.7.6	Range.....	51
2.2.7.7	An example Arduino code with Radio Head RH_RF95.h library.....	52
2.2.8	GSM and SMS messages.....	53
2.2.8.1	GSM circuits, modules and shields.....	53
2.2.8.3	Programming Arduino with GSM and SMS classes.....	54
2.2.8.4	An example of Arduino code with GSM.h and GSM_SMS.h libraries.....	56
2.2.8.5	Programming Arduino shield with SIM900 class.....	57
3.	Sensors and controllers (actuators).....	60
3.1	Simple actuators.....	60
3.2	Simple sensors.....	60
Note concerning the interruption enabled pins.....	61	
3.3	Motion detection sensors.....	61
3.3.1	PIR sensor.....	61
3.3.2	Microwave sensor – 10.525 GHz Doppler radar motion detector (RMD).....	63
3.3.2.1	An example of Arduino code operating with RMD module.....	64
3.3.3	IR sensor with transmitter.....	65
3.3.3.1	FC-51 sensor.....	65
3.3.3.2	An example of Arduino code for FC-51 sensor.....	66
3.4	Sensors (actuators) operating on I2C bus.....	67
3.4.1	An example of Arduino code for LM75 sensor.....	67
3.4.2	An example of Arduino code for HTU21D and BH1750 sensors.....	68
3.4.3	An example of Arduino code for AM2321 sensor.....	68
3.4.4	An example of Arduino code for BMP180 sensor.....	69
3.4.5	An example of Arduino code for DS3231 (Real Time Clock).....	70
3.4.6	OLED display operating on I2C bus.....	71
3.4.7	IMU (Inertia Measurement Unit) sensor.....	72
3.4.7.1	MPU 6050 sensor board.....	72
3.4.7.2	Two examples of Arduino code for MPU 6050 sensor.....	73
3.4.7.3	The execution results (fragment):.....	75
3.4.8	16-Bit ADC - 4 Channel with Programmable Gain Amplifier.....	76
3.4.9	Digital to Analog Conversion with PCF8591 module.....	77
3.4.9.1	PCF8591 YL-40 board.....	77
3.4.9.2	An example of Arduino code for PCF8591 YL-40 module.....	78
3.4.10	FM Radio Transmitter module.....	79
3.4.10.1	An example of Arduino code with FMTX library.....	79
3.4.11	FM Radio Receiver module.....	80
3.4.11.1	An example of Arduino code for TEA5767 module.....	81
3.4	Sensors operating on SPI bus.....	82
3.4.1	RFID sensor on 13.56 MHz.....	82
3.4.1.1	RFID 3.56 MHz ISM Frequency band.....	82

3.4.1.1	An example of Arduino code operating on RC522.....	82
3.4.2	SD card reader.....	84
3.4.2.1	The SD library.....	84
3.4.2.2	An example of Arduino code for micro SD reader.....	87
3.4.3	ArduCam (2MP) operating on I2C and SPI.....	88
3.4.3.1	ArduCam I2C bus operation.....	88
3.4.3.2	ArduCam SPI bus operation.....	88
3.4.4	MP3 Arduino shield with SD card and SPI.....	94
3.4.4.1	An example of programming MP3 shield with Arduino code.....	94
3.5	Serial – UART (tty) bus for sensors - actuators.....	96
3.5.1	Fingerprint scanner.....	96
3.5.1.1	FPM10A Arduino code – enrollment.....	97
3.5.1.2	FPM10A Arduino code – detection (not complete).....	98
3.5.2	MP3 player modules.....	100
3.5.2.1	DFPlayer Mini.....	100
3.5.2.1	An example of Arduino code for DFPlayer.....	100
3.5.2.2	CATALEX serial mp3 player.....	101
3.5.3	GPS.....	105
3.5.3.1	GPS working principle.....	105
3.5.3.2	NEO6M GPS module.....	106
3.5.3.3	An example of Arduino code to test NEO6M GPS module.....	106
3.6	Serial specific protocols on simple logic links (DIO).....	109
3.6.1	DHT11/22 sensors.....	109
3.6.1.1	An example of Arduino code for DHT11 and DHT22 sensors.....	109
3.6.2	7 segment LED display – TM1637.....	111
3.6.2.1	An example of Arduino code for TM1673 4-digit display.....	111
3.6.3	Ultrasonic sensors.....	112
3.6.3.1	An example of Arduino code for HC-SR04 ultrasonic sensor.....	113
3.7	Simple sensors with analog outputs.....	114
3.7.1	Current measuring sensor.....	114
3.7.1.1	Arduino code for DC current measuring (mode 5 A).....	114
3.7.1.2	Arduino code for AC current measuring (5A mode).....	115
4.	MCU and SBC boards – GPIO pins and buses.....	116
Introduction.....		116
4.1	The micro-controllers.....	116
4.1.1	ATMega328.....	116
4.1.2	ATMega2560.....	116
4.1.3	Xtensa LX106.....	116
4.2	Arduino Uno board.....	117
4.2.1	UART – RxD and TxD (D0 and D1).....	117
4.2.1.1	An example of Arduino code with SoftwareSerial.....	117
4.2.2	Interruption inputs – pins (D2 and D3).....	118
4.2.3	I2C bus pins.....	118
4.2.4	SPI bus pins.....	119
4.3	Arduino Nano board.....	119
4.4	Arduino Mini Pro (3.3V or 5V) board.....	120
4.5	Arduino Mega board.....	120
4.5.1	Arduino Mega pins and buses.....	121
4.5.1.2	SPI bus.....	122
4.5.1.3	I2C bus.....	122
4.6	ESP-12 (8266) based nodeMCU boards.....	123
4.6.1	ESP-12 NodeMCU (V 0.9).....	123
4.6.1.1	UART.....	124
4.6.1.2	I2C.....	124
4.6.1.3	SPI.....	124
4.6.2	ESP-12 Wemos D1.....	124

4.6.3 ESP8266 in deep sleep mode.....	124
Hardware Configuration.....	125
4.6.4 Using ESP8266 (Wemos D1) with MQTT protocol.....	127
4.6.5 Using ESP8266 (Wemos D1) with MQTT protocol in deep sleep mode.....	129
4.6.5.1 The mqtt client code:.....	129
4.6.5 Using ESP8266 (Wemos D1) with WebSockets.....	130
4.6.5.1 Keeping the data in EEPROM when in deep sleep mode.....	130
4.6.6 Using App Inventor to build simple application with ESP8266.....	133
4.6.6.1 The design editor and blocks editor.....	134
4.6.6.2 The Arduino code for Android applicationESP8266.....	135
4.7 SBC boards and GPIO.....	137
4.7.1 The RPI Hardware.....	137
4.7.1 The RPI Software.....	138
4.7.3 GPIO connections with wiringPi.....	138
4.7.3.1 wiringPi installation.....	138
4.7.3.2 Test wiringPi's installation.....	139
4.7.3.3 wiringPi - gpio utility.....	139
4.7.3.4 Simple logic outputs.....	140
4.7.3.5 Simple logic inputs.....	140
4.7.3.6 The RPI UART and wiringSerial.h.....	141
4.7.3.7 The RPI I2C: utilities and examples.....	143
4.7.3.8 The RPI SPI: utilities and examples.....	146
4.7.4 Interfacing RPI with Arduino and/or ESP boards.....	150
4.7.4.1 Interfacing via UART (/dev/ttyS0).....	150
4.7.4.2 Interfacing via I2C bus with Arduino Pro mini 3.3V.....	152
4.7.4.3 Interfacing via SPI bus in slave mode.....	153
4.7.5 Raspberry Pi and Long Range communication with LoRa (SPI bus).....	155
4.7.5.1 RFM95/96/97/98 circuits.....	155
4.7.5.2 Chistera-Pi board.....	155
4.7.5.3 LoRA GPS HAT.....	157
4.7.5.4 A simple extension board for RFM95/98 module.....	157
4.7.5.5 Simple RFM95/98 module and breakout.....	158
4.7.5.6 Simple sender receiver code.....	159
4.7.5.7 Wemos LoRa gateway communication with RPI LoRa module.....	162
4.7.6 Raspberry Pi as a WiFi Access Point with bridge to Ethernet.....	166
5. Examples of simple IoT architectures.....	169
5.1 First example: a simple Arduino Nano based architecture.....	169
5.1.1 The client front-end part (sensors).....	169
5.1.2 The client back-end part (network connection).....	170
5.1.3 The UDP server on RPI.....	172
To do:.....	173
5.2 Second example: A simple ESP8266 NodeMCU based architecture.....	174
5.2.1 Front-end of the server side.....	174
5.2.2 Back-end of the server side.....	175
5.2.3 The client side on SBC.....	177
5.2.4 To do.....	178
5.3 Third example: A simple communication system for IoT with Arduino Nano and NRF24L01 modules. .	179
5.3.1 Introduction to NRF24L01.....	179
5.3.2 NRF24 server.....	180
5.3.3 NRF24 client side with an RTC clock.....	181
5.3.4 To do.....	183
5.4 Fourth example: A simple communication system for IoT with Arduino Nano and LoRa SX1276.....	184
5.4.1 Simple experimental architecture LoRa technology.....	184
5.4.2 Server side (with selection of the clients).....	184
5.4.3 Client side with LM75 sensor.....	186
5.4.4 To do:.....	188

5.5 Fifth example: A simple communication system for IoT with Arduino Nano ESP8266 and LoRa SX1276 serial modems.....	189
3.5.1 Introduction.....	189
3.5.2 GPS to LoRa gateway (client).....	189
3.5.3 LoRa to WiFi gateway (server) on ESP-12 MCU.....	192
3.5.4 WiFi receiver on SBC.....	193
5.5.5 To do.....	194
5.6 Sixth example : sending images/streams from ESP to RPI3.....	195
5.6.1 Arduino code for ESP MCU.....	195
5.6.2 The HTML page on lighttpd server implemented on RPI3.....	199
5.6.3 To do.....	200
6. Building simple IoT gateways and WEB services.....	201
6.1 First project: a simple client-server system with an Arduino Nano as client and RPI3 as server.....	201
6.1.1 The client-side sub-system.....	201
6.1.2 The server side.....	203
6.1.3 To do.....	205
6.2 Second project: simple LoRa gateways (servers) for long range communication between tablets or smart-phones (clients).....	206
6.2.1 WiFi gateway code.....	206
6.2.2 To do.....	209
6.3 Third project: a gateway (client) with fingerprint scanner terminals for Internet communication with remote server.....	210
6.3.1 The enrollment of the users (scans).....	210
6.3.1.1 The enrollment program.....	210
6.3.2 The client side of the scanner.....	213
6.3.3 The server side.....	219
6.3.3 To do.....	223
6.4 Fourth project: an RPI3 based gateway with RF24 terminals.....	224
6.4.1 The server side.....	224
6.4.2 The client side.....	228
6.4.3 To do.....	230
6.5 Fifth project: an GSM (SMS) gateway with LoRa terminals.....	231
6.5.1 The gateway code.....	232
6.5.2 The terminal code.....	235
6.5.3 To do.....	237
6.6 Sixth project : an RPI3 gateway for ESP to thingspeak.com.....	238
6.6.1 Arduino terminal code with ThingSpeak.h library.....	238
6.6.2 Arduino terminal code without ThingSpeak.h library.....	240
6.6.3 To do.....	242
7. Projects – lists of components.....	243
7.1 Project 1.....	243
7.2 Project 2.....	244
7.2.1.Arduino code running on Wemos D1.....	244
7.2.2. C code running on RPI3.....	247
7.2.3 The second version of the project including remote configuration without external server.....	250
7.2.3.1 The gateway code on Wemos.....	250
7.2.3.1 The code running on the RPI3.....	256
7.3 Project 3.....	260
7.4 Project 4.....	261
7.5 Project 5.....	262
7.6 Project 6.....	263
8. RPI3 configuration.....	264
8.1 Network Configurations and softAP.....	264
8.1.1 Packages.....	264
8.1.2 Interfaces.....	264
8.1.3 hostapd configuration.....	265

8.1.4	dnsmasq configuration.....	265
8.1.5	Set up IPv4 forwarding.....	266
8.2	Installation and configuration for nRF24.....	267
8.2.1	Downloading and installing software.....	267
8.2.2	Connections and pin configuration.....	267
8.3	“socket” API for communication with Ethernet and WiFi.....	269
8.3.1	Working with UDP datagrams.....	269
8.3.1.1	IP/TCP “socket” address structure.....	269
8.3.1.2	Client-server message exchange application in datagram mode.....	270
8.3.1.3	Broadcasting.....	272
8.3.2	Working with TCP connections.....	273
8.3.2.1	A client/server TCP application – echo.....	273
8.3.2.3	Reading WEB page.....	278
8.3.3	Installing and Testing Mosquitto MQTT on the Raspberry Pi.....	280
8.3.3.1	JSON Data Payload.....	280
8.3.3.2	Installing Mosquitto.....	280
8.3.3.3	Configuring and Starting the Mosquitto Server.....	281
8.3.3.4	Testing the Mosquitto server.....	281
8.3.3.5	Using the Mosquitto RPI3 with an ESP on Wemos D1.....	281
8.3.4	Installing and Testing WebSocket server – tornado on the Raspberry Pi.....	282
8.3.4.1	Running the Tornado server with the Wemos D1 based client.....	283
8.4	Using Bluetooth communication with RPIs.....	284
	Introduction.....	284
8.4.1	Bluetooth transport protocols.....	284
8.4.1.1	RFCOMM.....	284
8.4.2.2	L2CAP.....	284
8.4.2	Installation.....	284
8.4.2.1	hciconfig.....	284
8.4.2.2	Python code on Raspberry Pi.....	285
8.4.3	Programming transmission with Bluetooth.....	286
8.4.3.1	Bluetooth stack Bluez.....	286
8.4.3.2	Bluetooth device address.....	287
8.4.3.3	RFCOMM socket address.....	287
8.4.3.4	L2CAP socket address.....	287
8.4.3.4	C Programming with files and I/O.....	288
8.4.3.5	The program – send and receive file with RFCOMM.....	288
8.4.3.6	The program – send and receive file with L2CAP.....	291
8.5	Using GSM modem to send/receive SMS messages.....	297
8.5.1	Basic C programming with AT commands.....	298
8.5.1.1	The SMS sender.....	298
8.5.1.2	The SMS receiver's.....	299
8.6	Using audio and video devices on RPI.....	301
8.6.1	Capturing and viewing images.....	301
8.6.2	Recording video with RPI camera.....	301
8.6.3	Using the RPI camera in Linux User Space.....	301
8.6.4	Using the USB webcams.....	302
8.6.5	Video4Linux2 utilities.....	303
8.6.6	Video4Linux2 programming.....	305
8.6.6.1	Capturing raw images.....	305
8.6.6.2	Capturing raw video.....	308
8.6.7	Streaming video.....	320
8.6.7.1	Streaming video from HD webcam.....	320
8.6.7.1	Streaming video from RPI camera.....	320
8.6.8	Audio on RPI.....	321
8.6.8.1	Essential audio tools.....	321
8.6.8.2	Internet radio playback.....	321

8.6.8.3 Recording audio.....	322
8.6.8.4 Bluetooth A2DP audio.....	323