Electronics

Rainbow Clock

User's manual

Document Revision. C

Table of contents

Glossary

1. Microcontroller

An embedded computer integrating a whole range of peripherals in a small package for convenience.

2. Bluetooth

A wireless communication standard permitting short-range exchange of data between electronic devices.

3. Refresh rate

A measure of the number of times an image can be drawn completely over the period of a second; this value is expressed in Hertz. Synonymous to *“FPS”*.

Product overview

1. Description

Rainbow Clock is an unusual timekeeping device characterized by an exotic look and designed with electronics in mind.

2. Capabilities

- Display the current time of the day
- Synchronize itself via a Bluetooth connection
- Alter its color scheme depending on events

3. Quick specs

- Microcontroller: PIC32MZ series
- LEDs: 60, RGB type
- Refresh rate: ~10Hz
- Power: 15Watt max, 0.5Watt minimal, 0.6Watt typical (measured on a prototype)

Basic operation

1. Reading the time of day

Each color encodes a distinct time unit. For each unit:

- 3 red adjacent LEDs represent the hours.
- 2 green adjacent LEDs represent the minutes.
- The remaining blue LED represents the seconds.

To ease reading, the clock's display is subdivided in four quadrants.

2. Changing the color scheme

Using the central dial, navigate the menu looking for "Color Config." Validate your choice. You will be presented with a choice of colors on the screen. Use the dial again, select a color and validate your choice.

3. Updating the time on the clock

Using the central dial, navigate the menu looking for "Time Config." Validate your choice.

Advanced operation

1. Synchronizing the clock using Bluetooth

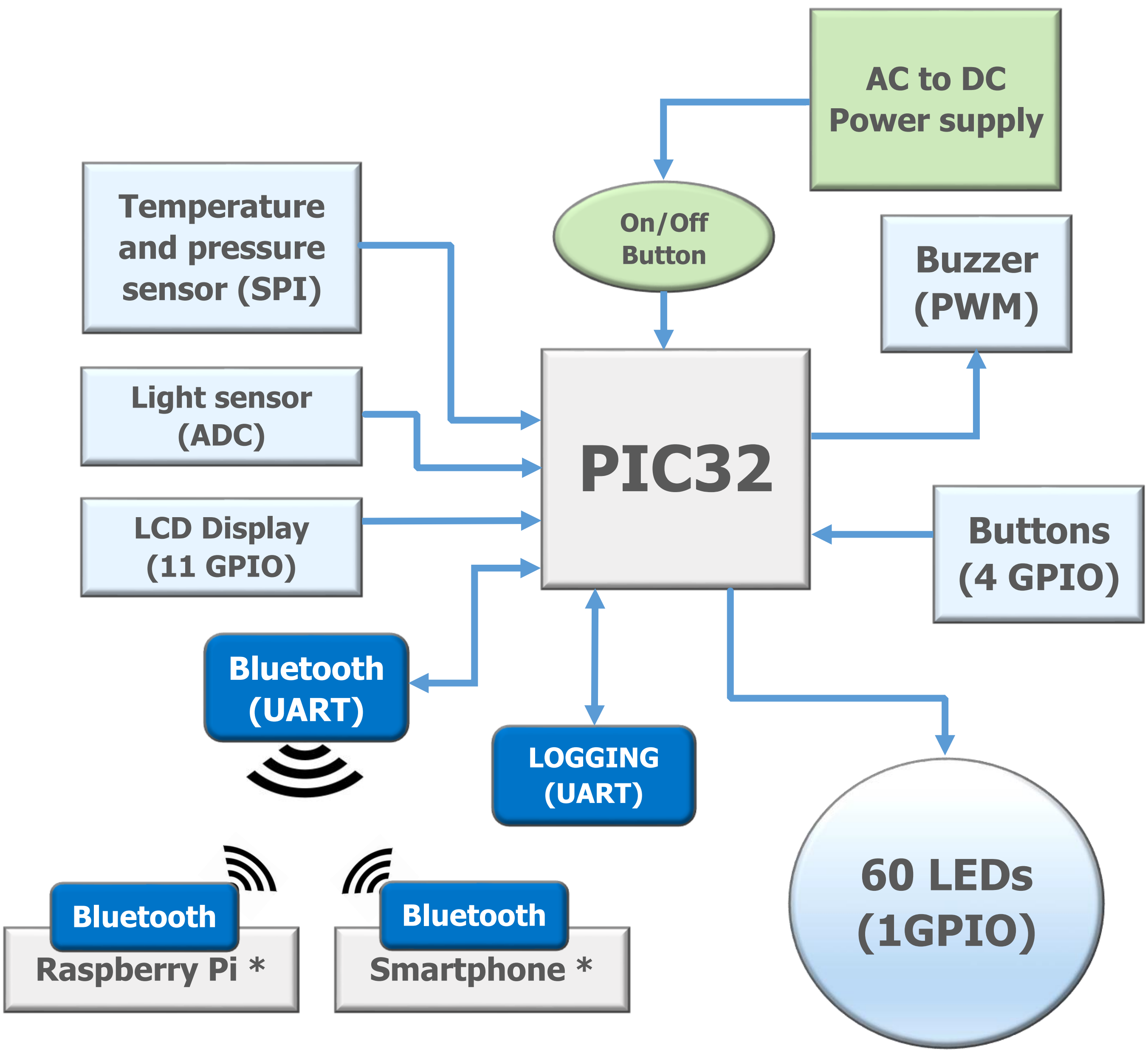
Initiate a serial connection to the Rainbow Clock’s Bluetooth module configured for a rate of 115200 bauds. The code to be sent is structured as follows:

“T**ss****mm****hh**ddMMYYYY”

With ‘T’ designating the Time config function.

Technical details

1. Block diagram



* Optional

2. Electrical consumption

- LEDs: Between 3.5V - 5.3V, about 60mA, 60mA x 60 = 3.6A max.
- PIC32: Between 2.3V - 3.6V, about 100mA.
- BMP280: Between 1.71V - 3.6V, about 4.2µA.
- RN4020: Between 3.0V – 3.6V, current is between 8mA 40mA.

  In line 3.3V the consumption of current is about 141mA, in line 5V current consumption is about 3.6mA maximum, the total consumption is 3.741A maximum.

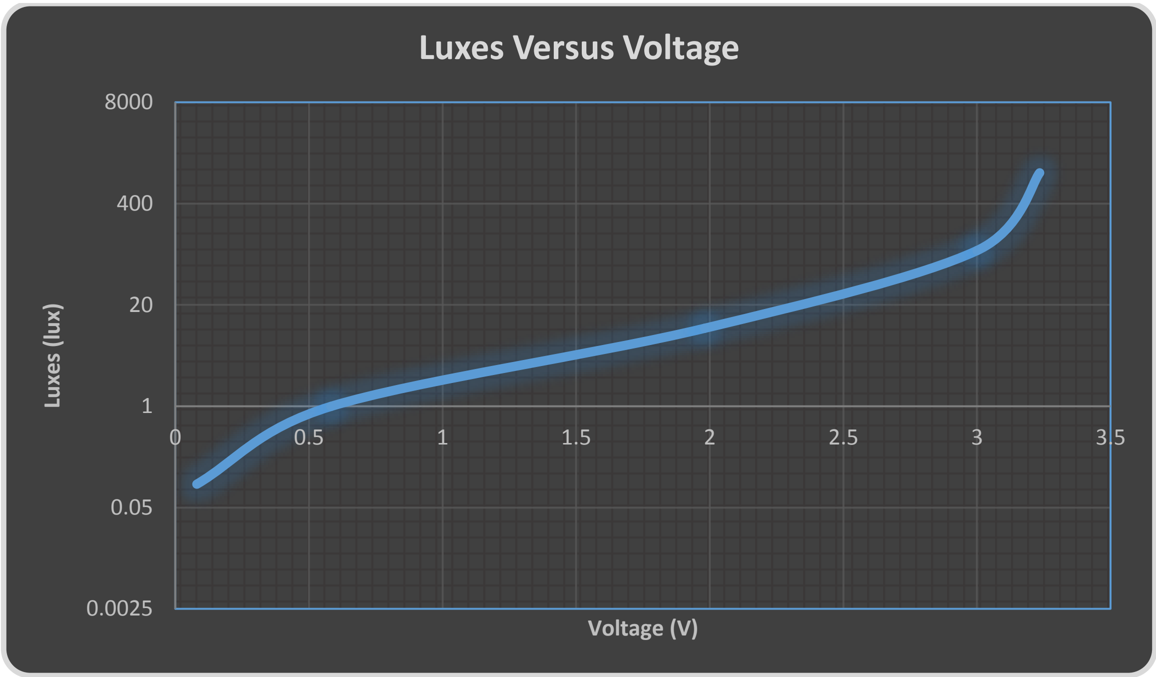
3. Components required for the project

•	1 *(one)*	PIC32xxxxxx microcontroller.	Ref: xxxxxxx
•	1 *(one)*	Strip of 60 RGB LEDs.	Ref fab: WS2812b
•	1 *(one)*	Incremental rotary encoder	Ref: 1191733
•	1 *(one)*	20x4 alphanumeric LCD screen	Ref: 2063162
•	x *(xxx)*	Resistors xΩ	Ref: xxxxxxx
•	x *(xxx)*	Capacitors xF	Ref: xxxxxxx
•	1 *(one)*	Voltage regulator	Ref: xxxxxxx
•	1 (one)	Digital pressure and temp sensor	Ref fab: BMP280
•	1 (one)	Bluetooth communication module	Ref: 2442930
•	1 (one)	5v Supply	Ref: xxxxxxx
•	1 (one)	Photocell	Ref: 7482280
•	1 (one)	buzzer	Ref: 2361105
•	2 (two)	push buttons	Ref: 1550267
	…		

We decided to use WS2812b as our LEDs for their convenience (They can be driven with a single wire using a simple protocol described in the datasheet of the component). (Not going to be bought from Farnell)

For our sensor, we chose the BMP280 or the BME280. It provides both temperature / pressure measurements and uses the I²C or SPI protocol. (Not going to be bought from Farnell).

A photoresistor will be used in a voltage divider setup, connected as an analog input pin.



We chose a rotary button for most of the interfacing with the clock. On the plus side, any watch user will figure out its multiples purposes.
Uses two GPIO plus two other GPIO buttons.

We are using a “Serial to Bluetooth” module for the device to communicate with smartphones and computers, so as to avoid the potential complexity of implementing a full Bluetooth stack, which we can’t afford at this time.

The chosen power supply can deliver up to 4A at 5v, and will still support the charge with all LED set at their maximum brightness level.

The chosen voltage regulator can deliver up to 1A in 3.3v for electronic modules using this amount of voltage.

4. Communications protocols used

1. UART

UART (or Serial) is a communication protocol, to use a same speed between PIC and Bluetooth we have choose 115200 bauds, in 8 bits mode, 1 Bit for stop, it will be used on Half Duplex mode.2.

2. I2C

The I2C protocol is essentially a bus made of 2 wires, SDA and SCL. SCL (serial clock line) is the clock speed for communication (the master imposes it), SDA (serial data line) is the line reserved for data transmissions.

I2C is a master-slave protocol. When a master starts to communicate, all others peripherals must stand by for instructions. The master first sends the starting bit, calling a slave peripheral by its address, before mentioning whether it wants to read or write data from it, then communicates the internal address of data.

For reading, the master sends SCL clock and reads back data sent by the slave peripheral via the SDA wire. If master is writing data in slave peripheral master will impose SCL clock and SDA data to write in slave. Of course each peripheral have a single addresses coded in 7 bits and each of them can be a master or slave.

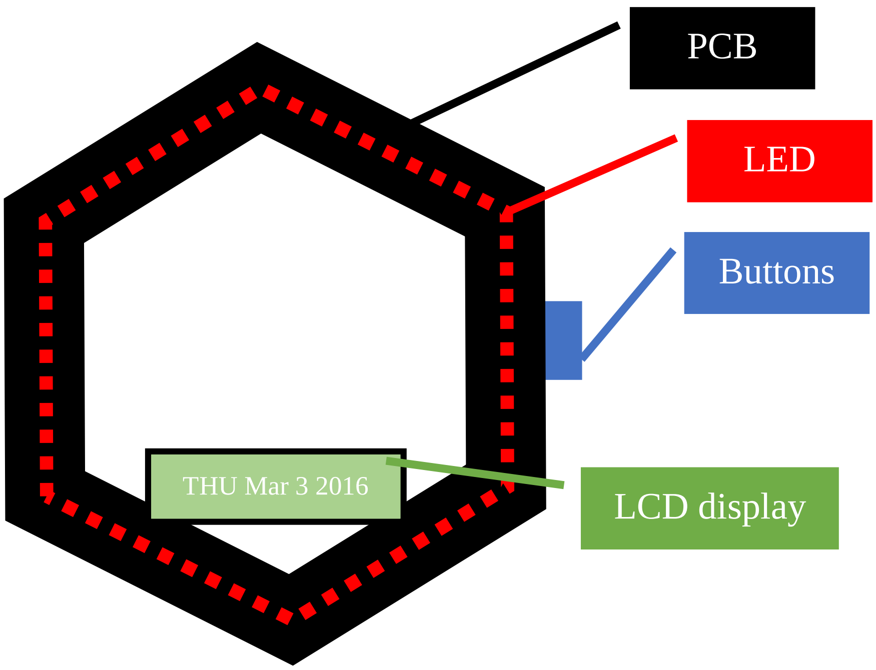
3. SPI

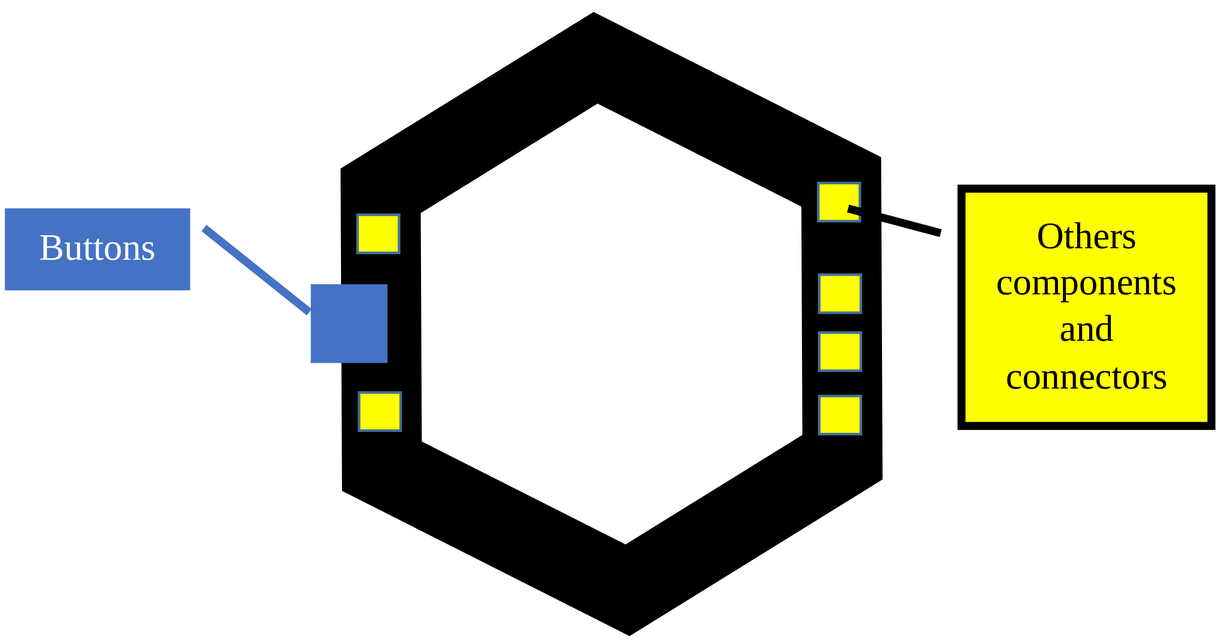
The SPI protocol is a communication bus composed by 4 wires, SCLK (clock rate speed transfer), MOSI (communication from master to slave), MISO (communication for slave to master, and SS (slave selection).

The master selects a slave with SS wire by connect the line in GND, and initialize the communication with slave, master send data with MOSI line and slave respond with MISO line, slave and master can communicate in same time (full duplex communication).

5. Implementation

We want to make the rainbow clock really different of other clock we imagine to make it hexagonal form in front all LED's and LCD display, in read all other components.





Planned features

- ***Display basic weather data and forecast using built-in sensors***

Data gathering could reveal itself being a nice addition to the project.

Contributions

In alphabetical order:

- **ltesson** ltesson@student.42.fr
- **nahmed-h** nahmed-h@student.42.fr
- **schiad** schiad@student.42.fr
- **vchesnea** vchesnea@student.42.fr

Page layout by: vchesnea

Endnotes