Weather Station Raspberry Pi project

Digital Microsystems Design

Professor

Marius Marcu

Students

Nashwan Azhari
Robert Krody
Tudor Vioreanu

Polytechnic University of Timisoara
Faculty of Automation and Computers
Information Technology and Computers, English spec.

Project summary

Our project consists of obtaining the environment temperature and humidity level, displaying them on an LCD and triggering appropriate LEDs.

In order to achieve this, we have used the following hardware:

- Raspberry Pi B+ model
- SHT11 Temperature and humidity sensor
- Shield LCD 16x2
- LEDs

Raspberry Pi B+ model specifications

Chip	Broadcom BCM2835 SoC
Core architecture	ARM11
CPU	700 MHz Low Power ARM1176JZFS Applications Processor
GPU	Dual Core VideoCore IV® Multimedia Co-Processor
Memory	512MB SDRAM
Dimensions	85 x 56 x 17mm
Power	Micro USB socket 5V, 2A

Connectors:

Ethernet	10/100 BaseT Ethernet socket
Video Output	HDMI (rev 1.3 & 1.4) Composite RCA (PAL and NTSC)
Audio Output	3.5mm jack, HDMI
USB	4 x USB 2.0 Connector
GPIO Connector	40-pin 2.54 mm (100 mil) expansion header: 2x20 strip Providing 27 GPIO pins as well as +3.3 V, +5 V and GND supply lines
Camera Connector	15-pin MIPI Camera Serial Interface (CSI-2)
JTAG	Not populated
Display Connector	Display Serial Interface (DSI) 15 way flat flex cable connector with two data lanes and a clock lane
Memory Card Slot	SDIO

Chip and memory

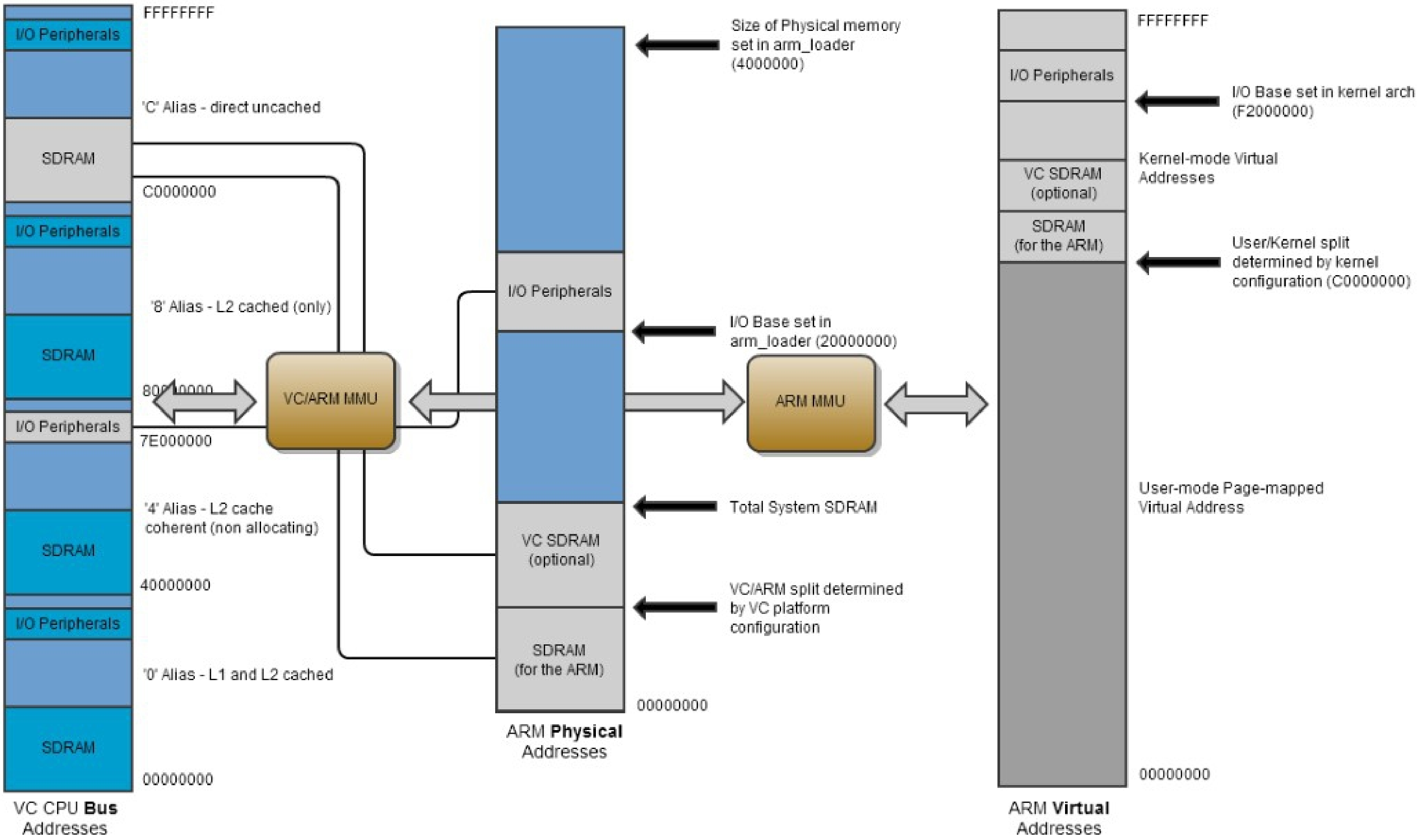
The Raspberry Pi uses a Broadcom BCM2835 system on a chip.

It features a VideoCore 4 GPU and a 700 MHz ARM1176JZF-S core (ARM11 family, ARMv6 instruction set).

The ARM core manages to provide a real world performance of 0.041 GFLOPS while the GPU is capable of 24 GFLOPS performance or 1 Gpixel/s in terms of graphics processing.

It has 512 MB of shared SDRAM (no dedicated video memory).

BCM2835 ARM Address map



FFFFFFFF
I/O Peripherals
'C' Alias - direct uncached
SDRAM
C0000000
I/O Peripherals
'8' Alias - L2 cached (only)
SDRAM
80000000
I/O Peripherals
7E000000
'4' Alias - L2 cache coherent (non allocating)
SDRAM
40000000
I/O Peripherals
'0' Alias - L1 and L2 cached
SDRAM
00000000
VC CPU Bus Addresses
VC/ARM MMU
I/O Peripherals
VC SDRAM (optional)
SDRAM (for the ARM)
00000000
ARM Physical Addresses
Size of Physical memory set in arm_loader (4000000)
I/O Base set in arm_loader (20000000)
Total System SDRAM
VC/ARM split determined by VC platform configuration
ARM MMU
FFFFFFFF
I/O Peripherals
VC SDRAM (optional)
SDRAM (for the ARM)
00000000
ARM Virtual Addresses
I/O Base set in kernel arch (F2000000)
Kernel-mode Virtual Addresses
User/Kernel split determined by kernel configuration (C0000000)
User-mode Page-mapped Virtual Address

Interfacing with the board

The SHT11 sensor, the display and the LEDs are connected using the 40 GPIO (General Purpose Input/Output) pins of the Raspberry Pi B+ Model.

These pins have kernel drivers for all major Operating Systems and pre-made libraries for multiple programming languages (BASIC, C/C++, Java, Perl, Python).

For our project, we have chosen Python and the RPi.GPIO CPython library and the Arch Linux OS for ARMv6 architecture.

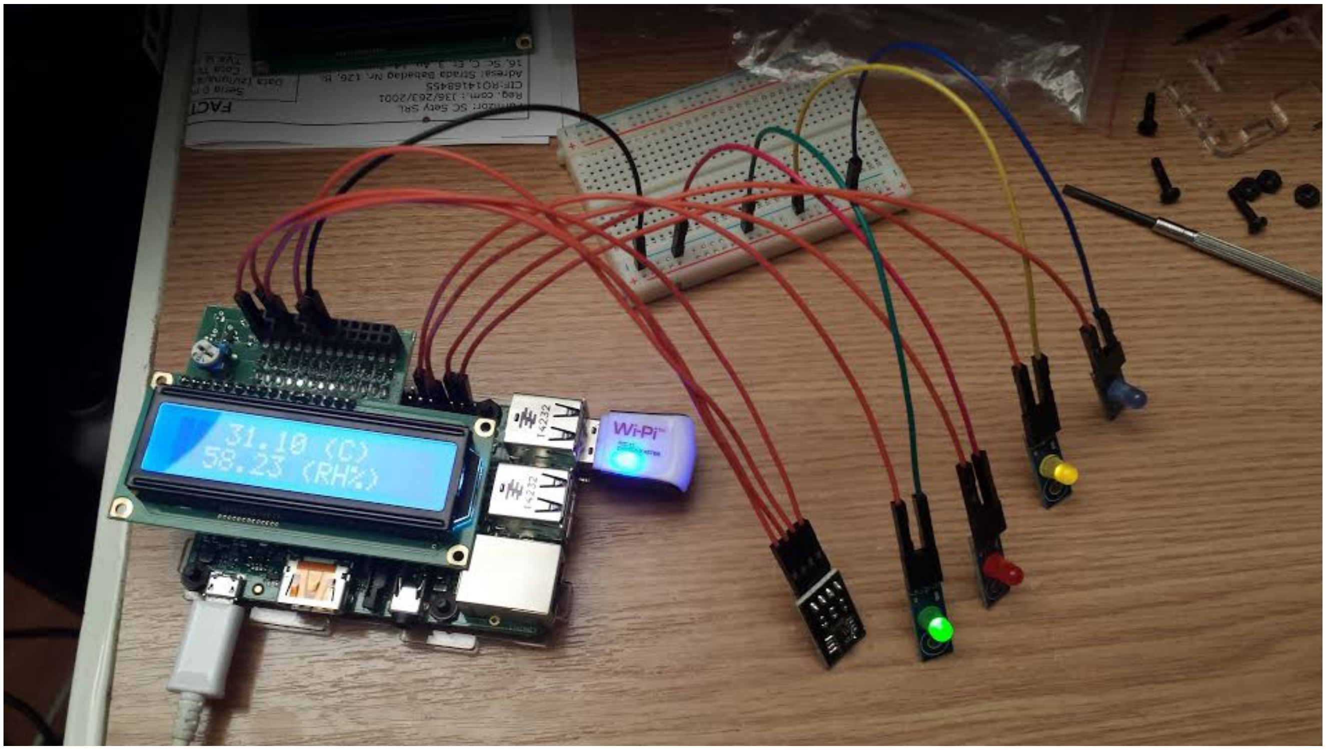
The LCD is connected using 6 pins in total: 1 mode selector, 1 serial clock and 4 data pins.

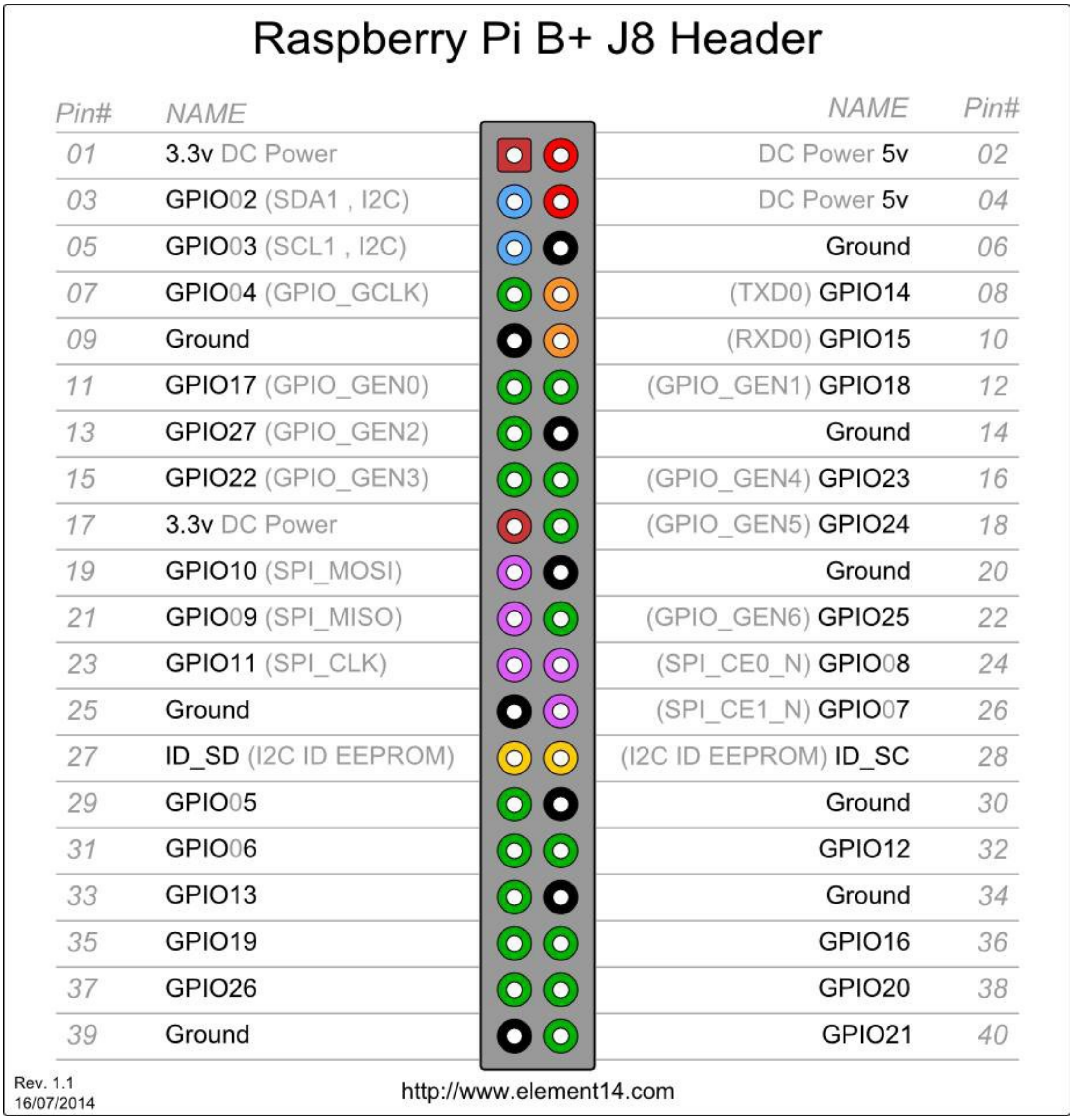
The mode pin is toggled between command (0) and data (1) and the serial clock is used to coordinate reading of the 4 bits from the 4 data pins.

The sensor is connected using a total of 4 pins: a Vcc pin, a ground pin, a data pin and a serial clock pin.

A command of 1 byte, whether 0x03 for temperature or 0x05 for humidty is read 1 bit per cycle. The corresponding 2 byte result is then returned, again 1 bit per cycle, along with an additional CRC Checksum byte, the use of which is optional with us choosing to abandon it.

The LEDs are connected using 2 pins each: a ground pin and an “IN” pin connected to a GPIO. An LED is turned on and off by turning the corresponding GPIO on or off.





Raspberry Pi B+ J8 Header

Pin#	NAME	NAME	Pin#
01	3.3v DC Power	DC Power 5v	02
03	GPIO02 (SDA1 , I2C)	DC Power 5v	04
05	GPIO03 (SCL1 , I2C)	Ground	06
07	GPIO04 (GPIO_GCLK)	(TXD0) GPIO14	08
09	Ground	(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)	(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)	Ground	14
15	GPIO22 (GPIO_GEN3)	(GPIO_GEN4) GPIO23	16
17	3.3v DC Power	(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)	Ground	20
21	GPIO09 (SPI_MISO)	(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)	(SPI_CE0_N) GPIO08	24
25	Ground	(SPI_CE1_N) GPIO07	26
27	ID_SD (I2C ID EEPROM)	(I2C ID EEPROM) ID_SC	28
29	GPIO05	Ground	30
31	GPIO06	GPIO12	32
33	GPIO13	Ground	34
35	GPIO19	GPIO16	36
37	GPIO26	GPIO20	38
39	Ground	GPIO21	40

Rev. 1.1
16/07/2014

http://www.element14.com

Pin mapping

LCD pins

1 mode selector	GPIO25 or #22
1 serial clock pin	GPIO24 or #18
4 data pins	GPIO17 or #11
	GPIO22 or #15
	GPIO18 or #12
	GPIO23 or #16

Sensor pins

1 data pin	GPIO27 or #13
1 serial clock pin	GPIO04 or #7

LED pins

Red LED	GPIO19 or #35
Yellow LED	GPIO20 or #38
Blue LED	GPIO21 or #40
Green LED	GPIO12 or #32

How is it all coded

Each component has it's own Python class. The WeatherStation class initiates a Display object, a Sensor object, and 4 LED objects using the parameters from the example.conf file. It then queries the sensor for temperature and humidity, writes the returned values on the LCD and triggers the approapriate LEDs.

Bibliography

http://en.wikipedia.org/wiki/ARM11
http://en.wikipedia.org/wiki/Raspberry_Pi
http://elinux.org/RPi_Low-level_peripherals
http://www.raspberrypi.org/wp-content/uploads/2012/02/BCM2835-ARM-Peripherals.pdf
http://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/Humidity/Sensirion_Humidity_SHT1x_Datasheet_V5.pdf
http://www.adafruit.com/datasheets/TC1602A-01T.pdf
https://learn.adafruit.com/downloads/pdf/drive-a-16x2-lcd-directly-with-a-raspberry-pi.pdf

Github Link

https://github.com/aznashwan/pi-sense