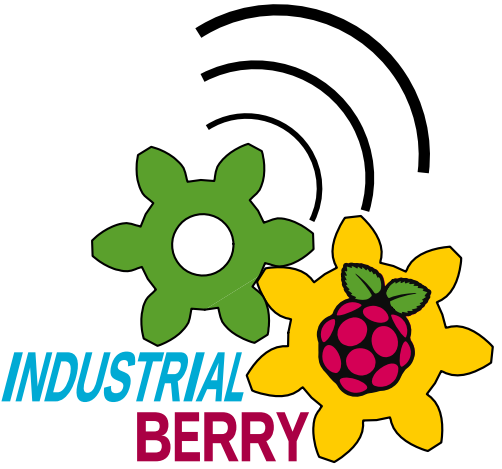
ADC Relay Adapter USB V 8.0 for Raspberry PI



www.industrialberry.com

May 2013

Contents

List of Figures

List of Tables

Chapter 1

License

Open-source hardware shares much of the principles and approach of free and open-source software. In particular, we believe that people should be able to study our hardware to understand how it works, make changes to it, and share those changes. To facilitate this, we release all of the original design files (Eagle CAD) for the IndustrialBerry hardware. These files are licensed under a Creative Commons Attribution Share-Alike license, which allows for both personal and commercial derivative works, as long as they credit IndustrialBerry and release their designs under the same license. The IndustrialBerry software/firmware is also open-source.

Chapter 2

Introduction

The ADC Relay Adapter V 8.0 (see Fig. 2.1) is an extension board for Rasp-Berry Pi based on Open Hardware Design. It has two principal hardware features: control of 4 relays and 4 ADC channels. The board is connected to the Raspberry with an USB Bus, this feature allow the compatibility with other linux boards.

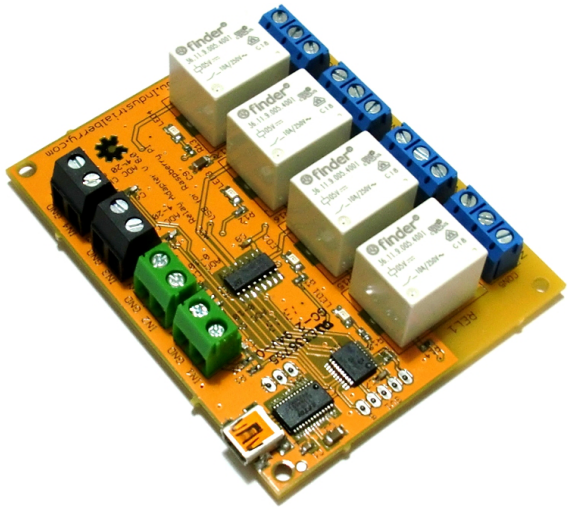


Figure 2.1: ADC Relay Adapter V 8.0

Chapter 3

Hardware implementation

The USB interface is implemented with a classical FT232RL [1] IC produced by FTDI. This IC is simple to use for the USB UART conversion. The IC3 a low cost PIC16F1829 [2], this MCU controls the 4 relays and manages the 4 ADC channels. We can see the follows pins connection:

- RC4 sends data to the Linux Board
- RC5 receives data from the Linux Board
- RC3 controls Relay 1
- RC6 controls Relay 2
- RC7 controls Relay 3
- RB7 controls Relay 4
- RA2 reads the ADC1 0-10V
- RC0 reads the ADC2 0-10V
- RC2 reads the ADC3 4-20mA
- RB4 reads the ADC4 4-20mA

We can see in figure 3.2 the part of the schematic with relays. Each device has a led for the status indication: Led Off -> Common Pin (C) connected with Normally Closed Pin (NC), Led On -> Common Pin (C) connected with Normally Open Pin (NO). The IC2 is the buffer ULN2003AD, this device can buffer 7 relays. Every channel contains two transistor in darlington configuration and diodes, see Fig.3.3.

The first two ADC channels can read input from 0 to 10 V, therefore they have to use a voltage divider with a ratio of about 5. This circuit is implemented with two resistors of 2kΩ and 8kΩ. The last two ADC channels can read input from 4 to 20 mA, this circuit is implemented with a resistor of 100Ω. With the maximum current of 20 mA the ADC reads a voltage value of about 2 V.

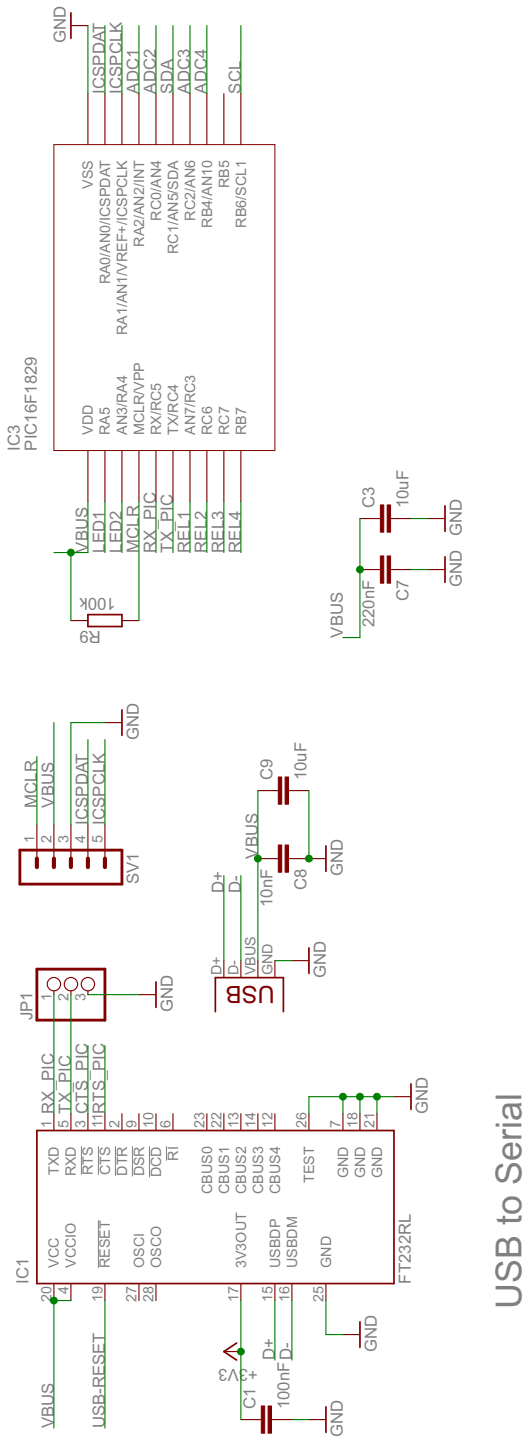


Figure 3.1: Schematic MCU USB

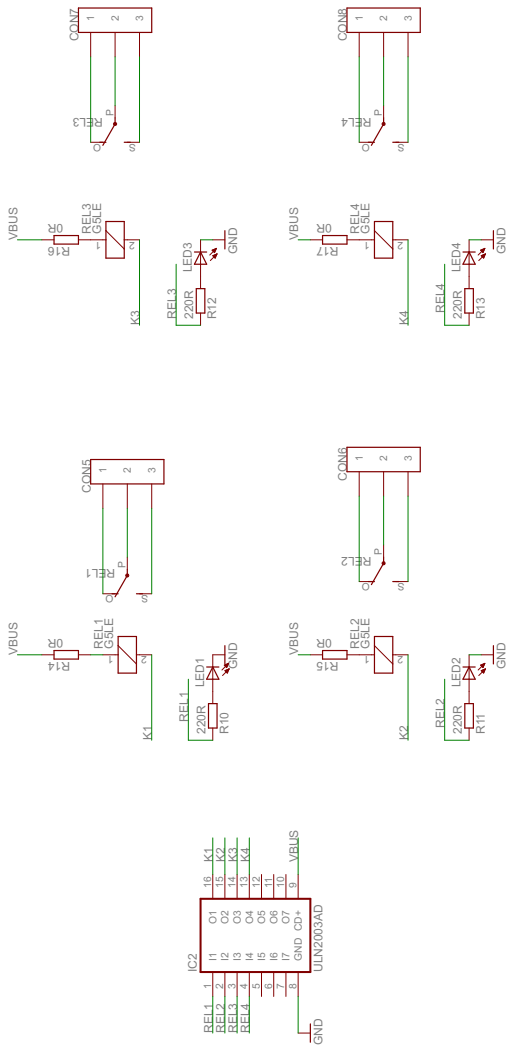


Figure 3.2: Schematic Relay

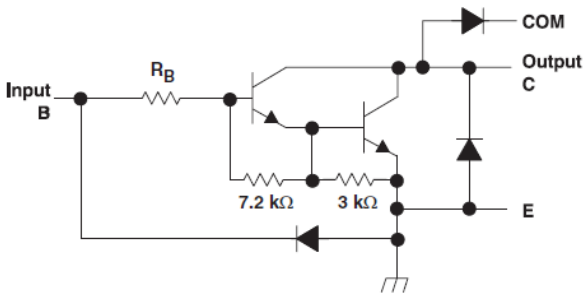


Figure 3.3: Channel of ULN2003AD

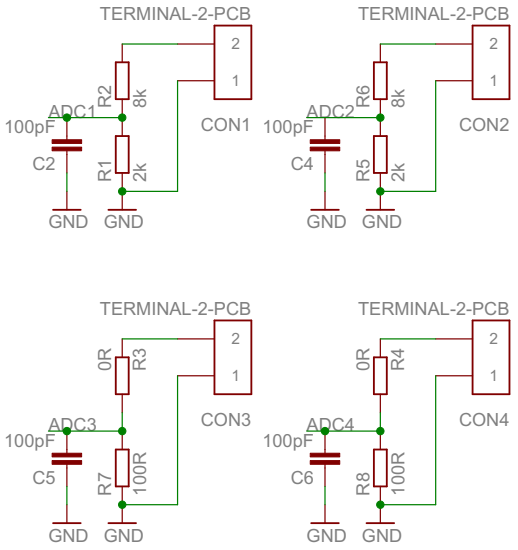


Figure 3.4: Schematic ADC Analog

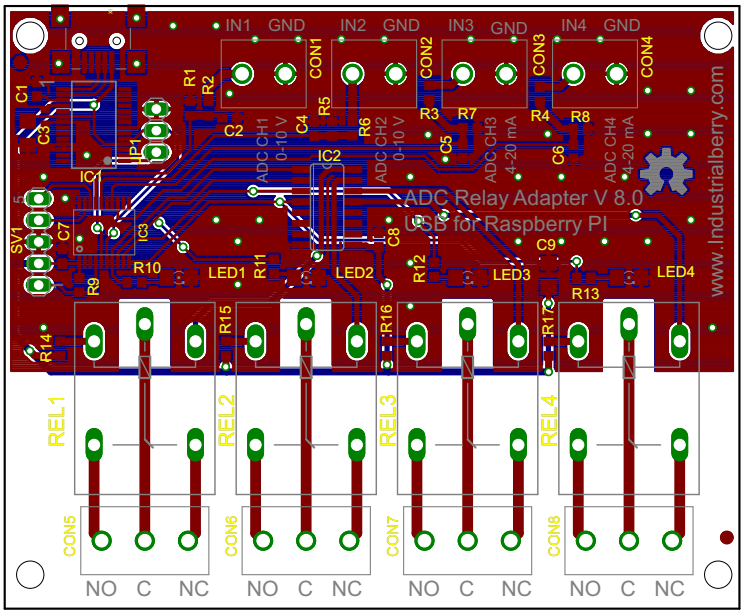


Figure 3.5: Board Layout

Chapter 4

Software implementation

The Raspberry Pi sees our board as a /dev/ttyUSB0. It is possible to control the board with a serial protocol from shell or a compiled program. This is the frame protocol:

- SOP (1 byte) Start of frame char 0x49
- CMD (1 byte) Command
- LEN (1 byte) Data length
- DATA (n byte) Data
- FCS (1 byte) Xor of all previous bytes

Possible commands:

- $SET_RELC_CMD = 0x02$
- $GET_ADC_CMD = 0x03$
- $GET_RELC_CMD = 0x04$
- $ADC_STATE = 0x05$
- $REL_STATE = 0x06$

The ADC Board produces an ACK = 0x00 for every correct message.

4.0.1 An example with the ADC

Get data from ADC:
We send this frame on ttyUSB0 ->
$SOP = 0x49$
$GET_ADC_CMD = 0x03$
$LEN = 0x00$
FCS

We receive from the board ->
The ACK message
SOP = 0x49
ACK = 0x00
LEN = 0x00
FCS

The ADC State
SOP = 0x49
$ADC_STATE = 0x05$
LEN = 0x08
DATA = $LO_UINT16(adc1)$
DATA = $HI_UINT16(adc1)$
DATA = $LO_UINT16(adc2)$
DATA = $HI_UINT16(adc2)$
DATA = $LO_UINT16(adc3)$
DATA = $HI_UINT16(adc3)$
DATA = $LO_UINT16(adc4)$
DATA = $HI_UINT16(adc4)$
FCS

4.0.2 An example with the Relays

Set the Relays:
We send this frame on ttyUSB0 ->
SOP = 0x49
$SET_RELC_MD = 0x02$
LEN = 0x02
$REL3_4 = 0x??$
$REL1_2 = 0x??$
FCS

We receive from the board ->
The ACK message
SOP = 0x49
ACK = 0x00
LEN = 0x00
FCS

The Relays State

SOP = 0x49
$REL_STATE = 0x06$
LEN = 0x04
$DATA = Status_Relay_1$
$DATA = Status_Relay_2$
$DATA = Status_Relay_3$
$DATA = Status_Relay_4$
FCS

Chapter 5

Components list

In the table 5.1 we can see the Bill of Material for the board, all the components are available on-line. For simplicity, every component has a DigiKey order code (www.digikey.com).

Quantity	Value	Package	Parts	Digikey-cod	Unit Price $
4	Yellow	1206	LED1, LED2, LED3, LED4	754-1144-1-ND	0.21
6	0 Ω	0603	R3, R4, R14, R15, R16, R17	P0.0GCT-ND	0.02
2	100 Ω	0603	R7, R8	RMCF0603FT100RCT-ND	0.04
4	220 Ω	0603	R10, R11, R12, R13	RMCF0603JT220RCT-ND	0.02
2	2 kΩ	0603	R1, R5	RMCF0603FT2K00CT-ND	0.04
2	8 kΩ	0603	R2, R6	P8.20KABCT-ND	0.11
1	100 kΩ	0603	R9	RMCF0603JT100KCT-ND	0.02
4	100pf	0603	C2, C4, C5, C6	490-4767-1-ND	0.10
1	10nf	0603	C8	445-1311-1-ND	0.10
1	100nf	0603	C1	445-1316-1-ND	0.10
1	220nf	0603	C7	445-1318-1-ND	0.10
2	10uf	1206	C3, C9	311-1376-1-ND	0.21
4	G5LE	G5LE	REL1, REL2, REL3, REL4	G5LE-1-36DC5-ND	1.45
1	FT232RL	SSOP-28	IC1	768-1007-1-ND	4.50
1	ULN2003AD	SSOP20	IC2	497-2345-1-ND	0.45
1	PIC16F1829	SSOP20	IC3	PIC16F1829-I/SS-ND	1.86
1	USBSMD	USBSMD	X1	WM17116CT-ND	1.59
4	TERMINAL-2-PCB	2 X 5.08mm	CON1, CON2, CON3, CON4	ED2609-ND	0.41
4	TERMINAL-3-PCB	3 X 5.08mm	CON5, CON6, CON7, CON8	ED2610-ND	0.45
1				PCB	4.00
				Total	24.20

Table 5.1: ADC Relay Adapter USB V 8.0 for Raspberry PI

Bibliography

[1] Future Technology Devices International Ltd. *FT232R USB UART IC Datasheet.*

[2] Microchip. *PIC16F1829 Datasheet.*