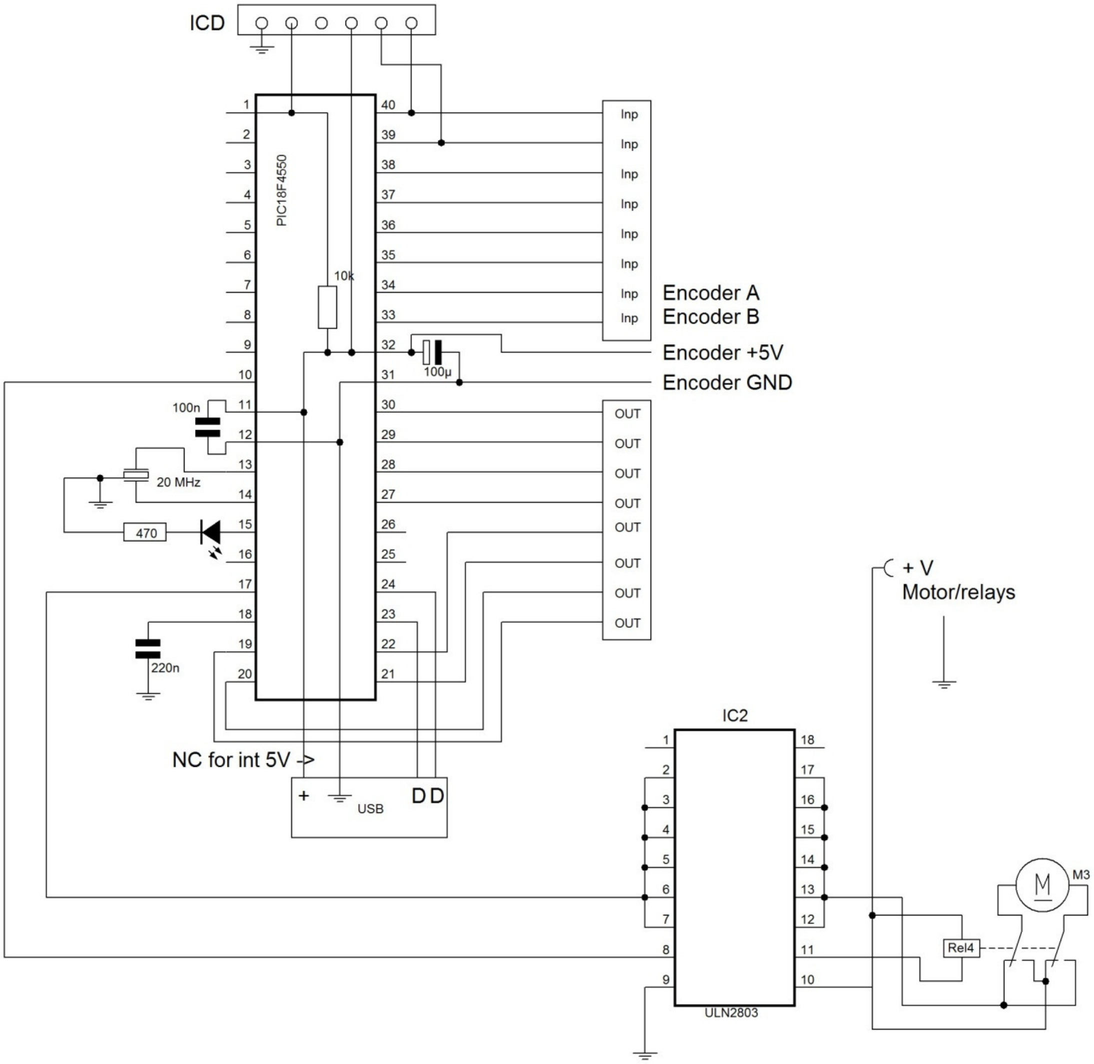
1. USB Rotary Table

1.1. Controller Board

The USB rotary table controller board can be built on a small breadboard . There is no need to design a circuit board, and the circuit used is very simple:



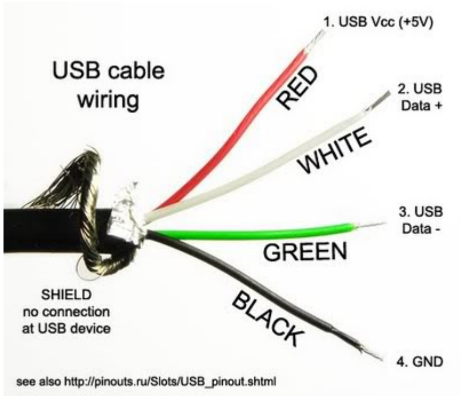
Only 2 ICs are used:

- PIC18F4550 (the pic microcontroller with USB port): this needs to be the exact type, as the software library (see below) expects this type of PIC. It is possible to work with another type, but then some programming changes are required.

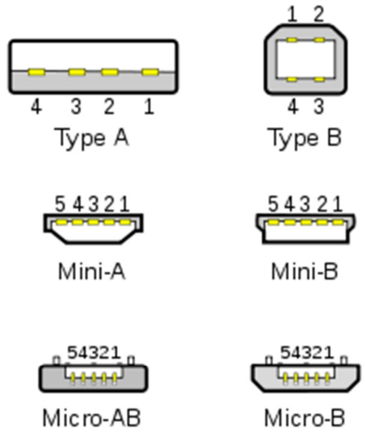
- ULN2803 (the output transistors for the motor and relay power): this can be any kind of power transistors/FETs that can be controlled by 5V TLL level, as this is absolutely not critical. Six of the internal darlingtons are paralleled for maximum motor output power

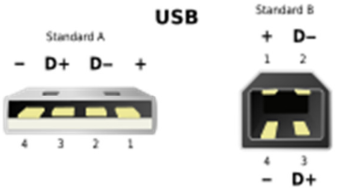
(2>7), and 1 darlington (8) is used for the direction switching relay (Rel4). I only used this IC because I had a small board available from a previous project.

- **USB connector** (for communication): you have 2 options here, or you solder a stripped USB cable directly to your board (pin 24=Data+ & pin 23=Data-):


USB cable wiring
1. USB Vcc (+5V)
RED
2. USB Data +
WHITE
3. USB Data -
GREEN
BLACK
4. GND
SHIELD no connection at USB device
see also http://pinouts.ru/Slots/USB_pinout.shtml

Or you wire an USB connector of some kind (pin 24=Data+ & pin 23=Data-):





Standard USB Pin-out & Cable Color Code
1)Red: V BUS (+5V)
2) White: D-
3) Green: D+
4) Black: Ground

Mini-USB Type-A Pin-out & Cable Color Code
1) Red: V BUS (+5V)
2) White: D-
3) Green: D+
4) Joined to pin 5 ID
5) Black: Ground

Mini-USB Type-B Pin-out & Cable Color Code
1) Red: V BUS (+5V)
2) White: D-
3) Green: D+

4) Not connected (*)ID
5) Black: Ground

(*) Sometimes joined to pin 5 via a resistor

- **ICD connector** (in circuit debugger): this is only necessary if you want to program the PIC when installed in the circuit board (necessary for me while developing). You can omit the connector if you do not need this.

- **Input connector** (PORT B of the PIC): only the first 2 inputs are used for the encoder signals. The other can be used for future extensions. Warning: as input are not buffered, be careful what you connect to them to avoid damaging the PIC

- **Output connector** (PORT D of the PIC): this can be used for future exertions, and does not need to be wired.

- **PIC/Encoder Power:** power of the PIC and the encoder can be supplied via the USB cable (as displayed in circuit diagram, or via a separate 5V power supply. My board is powered from the USB bus, and this works perfectly, so I see no need to build a second separate power supply.

- **Motor/Relay Power:** for this a separate power supply is needed, depending on the motor and relay used. Power to the motor is PWM controlled. For my rotary table, this power supply is 12Vdc. Remark: as a motor for a rotary table draws too much current, it is not possible to power the motor from the USB 5V, and I would also be wary of spikes generated on the power supply from switching the relay.

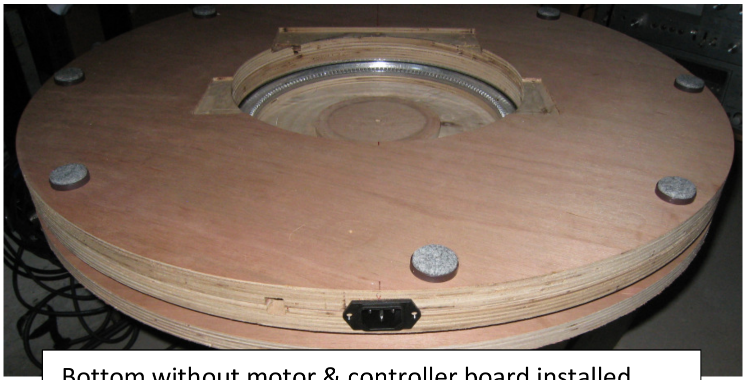
- **Motor Relay:** this relay is used to switch the rotating direction of the motor. Therefore, this relay minimum need two single pole double throw contacts (for example a DPDT relay). Please ensure the contacts are “break before make” to avoid short circuits when changing direction. Remark: my relay coil has an internal flyback diode, it is recommended to add this diode separately if the relay does not have it.

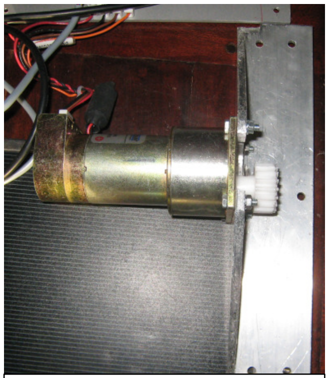
1.1.1.1. Motor + Encoder

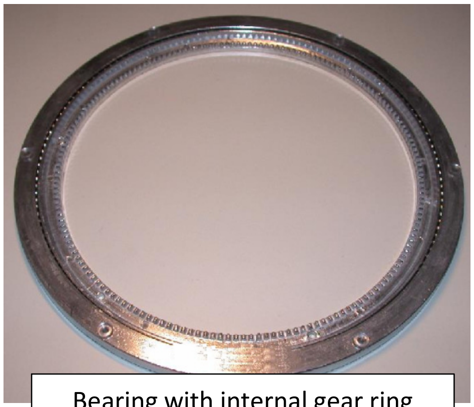
As the requirements for the motor mainly depend on how the rotary table is constructed, the load and the mechanical gearing, it is difficult to give precise specifications for the motor. I can however describe what I have done, and this can be used as a general guideline.

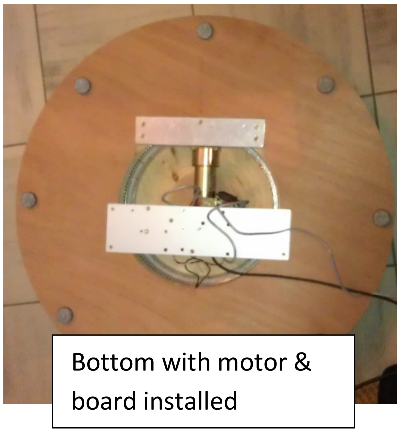
My USB rotary table looks like this:

Finished Rotary Table (without speaker stand)


Bottom without motor & controller board installed


Encoder + Motor + Gear


Bearing with internal gear ring


Bottom with motor & board installed

Used components:

- Motor+Gear+ Encoder: Pittman GM9234E496 (bought on Ebay for ±30€)
  - Motor torque (without gearbox): 43.1 x 10-3 Nm
  - Motor torque (with gearbox): **1.24 Nm**
  - Motor speed (no load with gearbox) : **94 RPM** (=> motor @5600 RPM)
  - Gearbox: 1/65.5
  - Encoder: 256p/rev
- Bearing with internal gear ring (bought on Ebay for ±40€)
  - Search on www.ebay.de for “Drehkranz Drehlager”, is supplied by several members
- Multi-plex wood table with a diameter of 70cm (± 15€)
- PIC micro controller board (see above) (±25€)

Although the used motor is a 24Vdc model, I use it with a 12Vdc power supply, because the PIC (or more likely my code in the PIC) is not fast enough to cope with the motor running @5600 RPM with the fitted 256p encoder… Running the motor at 12Vdc I can run at full speed without losing pulses. Even at 12Vdc, the motor power is strong enough to move the table with me sitting on top (>80kg…)

Conclusions for motor selection:

- Try to select a encoder with **less than 256p/rev** to run the motor at full speed
- Try to select an encoder with TLL signal levels if you want to connect it directly to the PIC inputs as I did
- Try to select a gearbox with **a gear >65.5** (preferable 100 – 150 when running the motor at full speed)
- Try to have an **output torque of > 1 Nm** (motor with gear)
- Don't spend too much money on it, every now and then some nice models appear on Ebay
  - Brands to look for: Pittman, Maxon, Faulhaber, Globe Motors
- Also take a look at some DIY robotics sites, they have useful motors also …

1.2. PIC Source Code

The USB rotary table was based on the excellent "Open Source Framework for USB Generic HID devices" from Simon Inns. As he has an excellent website where everything is described into great detail, I think there is no need to repeat this here, but I recommend the user to visit the following webpages:

- http://www.waitingforfriday.com/index.php/Open_Source_Framework_for_USB_Generic_HID_devices_based_on_the_PIC18F_and_Windows
- http://www.waitingforfriday.com/index.php/Building_a_PIC18F_USB_device

Again many thanks for this framework, without it, I wouldn't have been able to write this application.