HOW TO MOVE YOUR WHEELCHAIR WITHOUT TOUCHING IT

A guide to retrofitting a Raspberry Pi to an off-the-shelf electric wheelchair

Lewis Brand
University of Nottingham School of Computer Science

Contents

Safety warnings

- Please read your manufacturer's documentation before using your wheelchair or undertaking any modifications.

- There are steps in this guide that deliberately change or bypass safety features of the wheelchair. Please read this guide fully and ensure you understand the modifications and the consequences of these modifications before undertaking these steps.

- The wheelchair is heavy and powerful. Please ensure the drive wheels are removed or not in contact with the ground whilst testing the wheelchair.

- Electric wheelchairs have a battery, typically lead acid, which can cause electric shock. Make sure to isolate this power supply before working on the wheelchair. Keep tools and other metal objects away from the battery terminals.

- When you are not using your wheelchair, disconnect any control systems that have been retrofitted to prevent battery discharge.

- The Penny and Giles GC2 Joystick controller is susceptible to interference from electromagnetic and radio frequency sources. In some cases, this can cause unintentional movement of the wheelchair. Please avoid using the wheelchair near sources of electromagnetic and radio frequency interference.

- Ensure you are wearing suitable PPE when working with tools.

Disclaimer

This guide is designed around the Jazzy Select 6 by Pride Mobility with the PG GC2 controller and is correct as of April 2022. Your wheelchair may be different. Please take the material within this guide as guidance only and refer to your manufacturer's documentation before attempting any modifications suggested.

Material within this guide may refer to actions that are advised against or expressly prohibited by the manufacturer. Undertaking any modifications may void the manufacturer's warranty on your wheelchair as well as any safety or medical certification that the wheelchair may hold.

Whilst every effort is made to ensure the accuracy of the information in this guide, it is provided "as-is" and the author and its associates make no representation or warranties of any kind, express or implied about the completeness, accuracy, reliability, suitability, or availability with respect to the guide or the information, products or related graphics contained in this guide for any purpose. The author and its associates accept no liability or responsibility to any person as a consequence of any reliance upon the material contained in this guide. Under no circumstances, including negligence, shall anyone involved in creating or maintaining this guide be liable for any direct, indirect, incidental, special or consequential damages, or loss that result from the use or inability to use this guide. Any reliance you place on this guide is strictly at your own risk.

About the project

Powered wheelchairs are an excellent tool helping to restore mobility to people who cannot use normal wheelchairs and allowing them to work, attend school or college, and have a social life. Unfortunately, these wheelchairs are far from perfect with many people having difficulties in controlling them in crowded areas, when moving backwards, and in tight spaces. Smart wheelchairs try to solve this problem with features like automatic route planning and object avoidance, but these are so expensive that they are out of reach from the people who would benefit most from them.

That is where this guide comes in. We wanted to find a way to help anyone with a bit of technical knowledge and some soldering skills to convert a basic electric wheelchair to a fancy all singing all dancing, mobility machine! The goal of this project was to use readily available components to retrofit object detection and avoidance to an off-the-shelf electric wheelchair. This guide will talk you through how a standard electric wheelchair works, what each part does, and explain how to add a small computer to it so you can control it.

This guide is targeted at people with good dexterity, through-hole soldering skills, basic electronics knowledge, and moderate computing knowledge. All the parts in this guide should be easy to obtain and require little effort in assembling.

What you will need

This guide is designed around the Jazzy Select 6 by Pride Mobility with the PG GC2 controller. The following parts list is based around these so you may need to adjust these depending on your wheelchair.

Tools

- Soldering iron
- Side cutters
- Needle nose pliers (optional but useful)
- Crosshead screwdriver
- Cutting board
- Heat gun
- Permanent marker
- Wire strippers
- Small drill/rotary tool
- Allen keys
- Heat-shrink tubing
- Craft knife
- Heat proof mat

Parts list

- Raspberry Pi 3B+ (or newer) and microSD card preconfigured with Raspbian and SSH access
    - The 3B+ will have enough power to run the ROS operating systems and read in many sensors at the same time
- Adafruit ADS1015 4 channel analogue to digital converter (ADC)
- Adafruit MCP4728 4 channel digital to analogue converter (DAC)
    - If you are using different ADCs and DACs, make sure they accept a 5V input voltage and support I^2C connections.
- DC-DC Buck converter accepting an input of 25V and outputting 5V
- Proto strip board for Raspberry Pi
- We recommend one with a similar layout to a breadboard for easy use
- General-purpose IP66 enclosure
    - We recommend a minimum of 120x160x75mm
- 3x RJ45 breakout boards
- 2x panel mount RJ45 sockets
- 2x panel mount USB sockets
- Solid core wire 0.64mm, 22 AWG
    - Or male to male breadboard jumper wires
- 2x Ethernet cables, one meter or longer
    - Ideally use distinct colours
    - Do not use crossover cables
    - Ethernet is easiest as we will not be using any of the shielding which is present in Cat6 and above
- Normally closed emergency-stop button (optional but advised)
- Twin core 0.64mm, 22 AWG wire (optional for stop button)
- Small thin cable ties
- Large long cable ties
- 4x adhesive PCB standoffs

Wheelchair overview

The Jazzy Select 6 is formed of two main parts: The seat assembly and the power base assembly. The seat assembly includes two armrests, a seat back and a seat base.

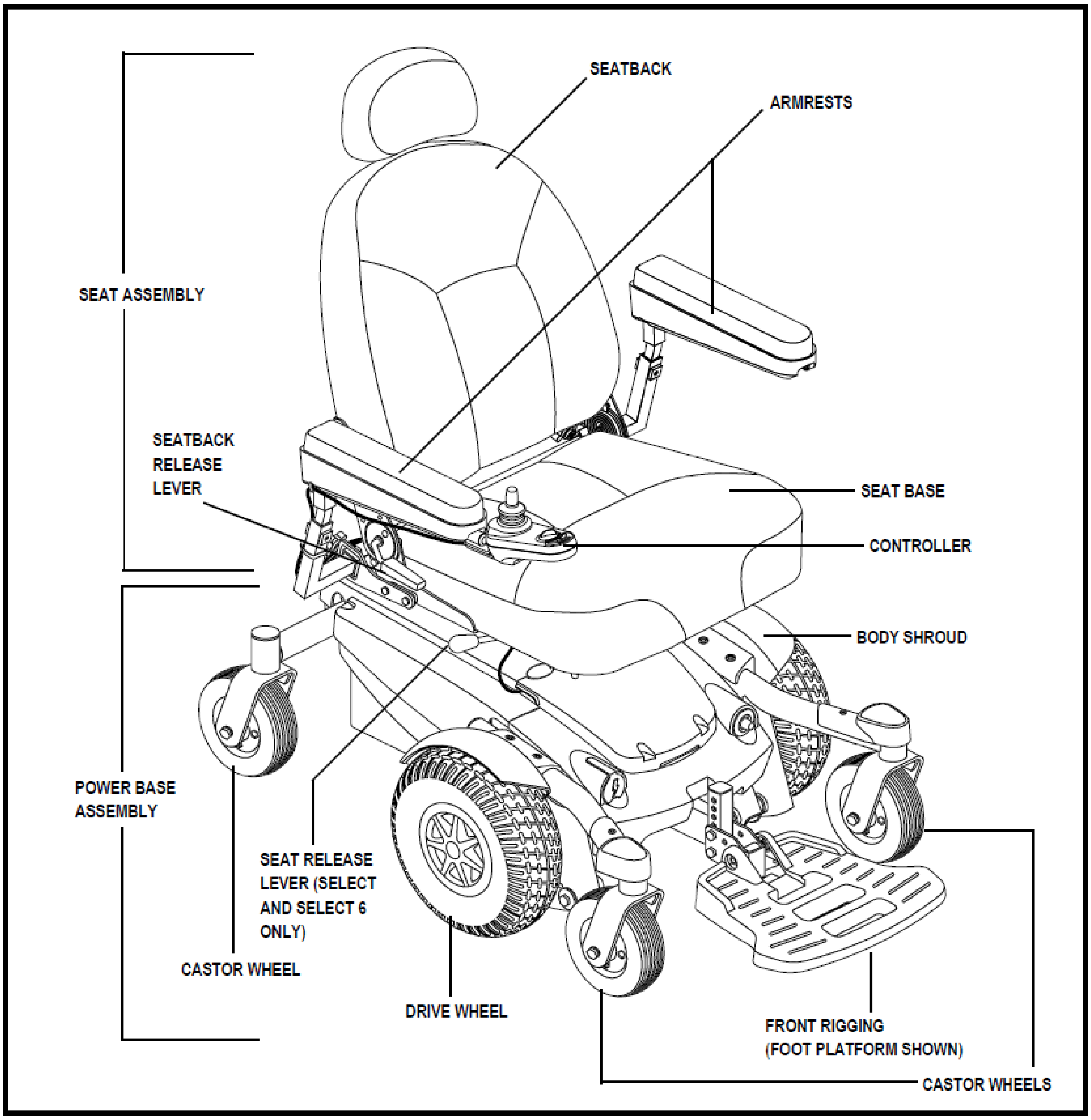


Figure 1 The Jazzy Select Series Power Chair [1]

The PG GC2 controller is normally mounted on an attachment pole that connects on to the underside of either armrest.

The power base assembly houses two batteries, the main circuit breaker, the motor controller, two motor/brake assemblies, and the wiring harness. It also includes two drive wheels towards the centre of the wheelchair, two rear castor wheels and two front anti-tip wheels.

The two rear castor wheels are mounted on a pole at the back of the wheelchair. This pole is attached to a pivot point at the centre of the wheelchair allowing the castor wheels to move up and down. As the left wheel moves up, the right wheel will move down and vice versa.

The front anti-tip wheels are mounted on poles either side of the wheelchair. Each side is independent of the other. The pole to which the anti-tip wheel is attached is also attached to the motor/brake assembly on that side. This is collectively mounted to the wheelchair on a pivot point between the anti-tip wheel and the drive wheel. This means that as the anti-tip wheel goes up, the drive wheel on that side will go down slightly due to where the pivot point is.

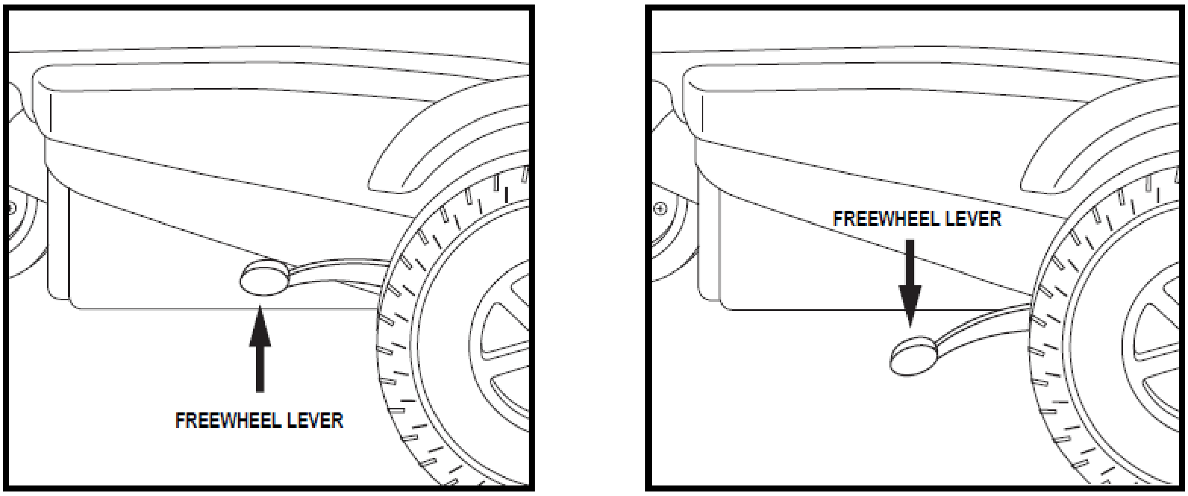


Figure 2 Drive mode engaged on the left and drive mode disengaged on the right [1]

The Jazzy Select 6 has a manual freewheel lever on each motor which allows you to disengage the motors from the gearboxes and manoeuvre the chair manually. Disengaging the drive also disengages the brake. Each motor has a freewheel lever which when pushed up, engages the motor and brake. When pushed down, the motor and brake are disengaged, and the wheelchair can be moved freely.

The control harness connects the motor controller in the power base to the joystick controller of your choice. The Jazzy Select 6 includes a PG GC2 controller and uses a four-pin water resistant connector.

A note on operating behaviour

There are two points worthy of note here related to how the wheelchair behaves.

When either of the drives are disengaged (when freewheel is active), the PG GC2 controller will flash an error and the wheelchair cannot be controlled.

The wheelchair has pre-set acceleration curves which differ depending on the direction of travel. In any direction, the wheelchair will start slowly and ramp up to the desired speed. When reversing or turning in the wheelchair, a speed cap is applied. When moving forwards, this speed cap is much higher. The speed controls on the GC2 adjust how fast the wheelchair moves forwards and how fast the wheelchair turns (although to a lesser extent). It has little effect on reversing speed. The wheelchair is reprogrammable with a proprietary programmer, and it may be possible to change these parameters, but this has not been tested.

Whenever the wheelchair is stopped, the parking brake is automatically applied. An audible clunk can be heard when this happens.

The PG GC2 controller

The Penny and Giles GC2 controller is simple, cheap, and very commonly used on a wide variety of powered wheelchairs. The GC2 has a 5 LED display, power button, horn button, two axis joystick, speed control dial, 3-pin XLR charging port, and cable that links to the control harness.

Figure 3 The front and inside of the PG GC2 joystick controller

The control harness has four colour coded cores. The red core is 25V to power the GC2, the black core is 0V, the white core is a communications cable between the GC2 controller and the motor controller, and the yellow core is a battery charging cable.

The yellow wire only goes to one of the pins on the charging port. The red and black wires are also connected to the charging port and then, with the white wire, goes via a ferrite ring to the internal circuit board.

The red, white, and black wire connect to this circuit board using a removable connector.

The 25V from the red and black wires is reduced on the circuit board using two voltage regulators to 5V. This 5V supply powers all the control circuitry on this circuit board as well as the joystick and the speed dial.

The speed dial is a 10KA potentiometer which takes the 5V (red wire) and a reference 0V (black wire) and supplies a centre tap (yellow wire) which is used by the control circuitry to set the speed of the wheelchair.

The joystick is a PG Drives D51422 dual axis hall effect joystick which is built in house by Penny and Giles. It uses a set of four analogue hall effect sensors to decide the position of the joystick, two on the X axis, and two on the Y axis. The joystick is connected to the circuit board with an 8-pin ribbon cable.

How to override the wheelchair

There are several ways to control how the wheelchair moves:

Method one is to bypass the motor controller. This method gives you direct control of the motors and brakes so you could (in theory) do whatever you wanted to. The downside of this method though is that you would need some beefy and expensive motor controllers; you would have to handle the batteries and add various over/under current protections and charging management etc. If you have the time, budget, and knowledge, this method will give the best results but is complicated and has a lot of safety risks. We strongly recommend against this method unless you have the experience and knowledge to work directly with the batteries and motors.

Method two is to pretend to be a GC2 controller. This means all the motor and battery management is still handled by the motor controller and you just supply the control signals to the motor controller. The problem with this method however is the control protocol between the GC2 and the motor controller is proprietary and undocumented so you would have to do a lot of signal analysis first to reverse engineer the communications signal. This method gives you the same amount of control as method three but with even more work, so we recommend going with the third method.

Method three gives less control than method 1 but is much (much much) easier. Why go through all the work trying to decipher the control protocol or prevent batteries from exploding when you already have the circuitry that does it for you! Instead of pretending to be the entire GC2, this method involves pretending to be the just the joystick. The GC2 reads the signals that the joystick is outputting and translates them into control signals for the motor controller. By imitating the joystick, we can trick the GC2 into sending any command we need.

How the joystick controls the wheelchair

So, we know we want to pretend to be the joystick but to do this, we need to know exactly what it is doing.

As we mentioned earlier, the joystick used in the GC2 is a PG Drives D51422 dual axis hall effect joystick. It uses four analogue hall effect sensors to decide the position of the joystick, two on the X axis, and two on the Y axis. Each pair of hall effect sensors are set up in a way known as a “same-ramp” configuration and is a safety feature of the joystick. In a same-ramp configuration, as one sensor’s output increases, so should the other so if the difference between the signals in a pair of sensors gets too much, you know there is a fault with the joystick. If the GC2 detects a significant difference within a pair of sensors, it will stop the wheelchair, engage the parking brake, and display an error code. This is the reason why we need to send all the signal cables to our control unit and not just one from forward/backward and one from left/right.

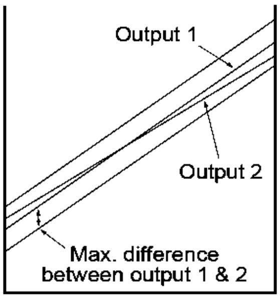
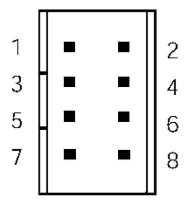


Figure 4 Same Ramp configuration graph [2]



Pin Number

XY Joystick

1 Positive voltage supply
2 Left/Right output 1
3 Zero voltage supply
4 Forward/Reverse output 1
5 Forward/Reverse output 2
6 Center tap
7 Left/Right output 2
8 Switch output (NC if no switch)

Figure 5 Ribbon cable pin layout and numbering [2]

The joystick is connected to the circuit board with an 8-pin ribbon cable. Pin 1 is the positive voltage supply (5V), pin 2 is the left/right output 1, pin 3 is the zero Volt supply (0V), pin 4 is the forward/reverse output 1, pin 5 is the forward/reverse output 2, pin 6 is a centre tap, pin 7 is the left/right output 2, and pin 8 is not connected. When the circuit board is viewed from the side of the power connector, pin 1 is the top right pin, pin 2 is beneath it, pin 3 is to the left of pin 1, pin 4 is to the left of pin 2 etc. Pin 8 is on the bottom left. The centre tap is used as a "zero position" reference voltage. This means when the joystick is in the centre, the voltage from all the hall effect sensors should match the centre tap.

When moved, the voltage output from the hall effect sensors changes. When pointing full forwards, the forward/back outputs show +1.5V from the centre tap. When pointing full backwards, the forward/back outputs show -1.5V from the centre tap. When pointing full right, the left/right outputs show +1.5V and when pointing full left, the left/right outputs show -1.5V.

If we want to just read the signals being output from the joystick, all you would need to do is tap onto the signal cables of the joystick and read their voltages. To physically override this signal however, we need to completely cut these cables to prevent the original joystick signal from interfering with our new control signal.

The Raspberry Pi

To control the wheelchair, we need a system that will be portable, be able to communicate with the wheelchair, and be able to process any sensors that you attach. The Raspberry Pi fits all these criteria. The Raspberry Pi is powerful enough to run ROS (the robot operating system) which will control the wheelchair and its networking capabilities means that if we want to do anything more complex, we can have it communicate easily to another machine. The Raspberry Pi is also small, light, and can easily be tucked into a wheelchair and consume little power when running so it will not drain the battery of the wheelchair quickly. The Raspberry Pi is also incredibly cheap and very accessible for all users from complete novices to advanced users with a vast number of guides and videos available.

We will use the GPIO of the Raspberry Pi to communicate with the joystick of the wheelchair. Unfortunately, the joystick controls the wheelchair by changing a voltage within a range. This makes it an analogue signal. The Raspberry Pi GPIO does not directly support analogue signals, so we need to find a way around this.

I^2C, ADCs, and DACs

The Raspberry Pi does not support directly reading analogue signals using the GPIO, but it does support I^2C (Inter-Integrated Circuit) which is a way for different electrical components to communicate over short distances. Using I^2C we can expand the Raspberry Pi to be able to input and output analogue signals. ADCs (analogue to digital converters) and DACs (digital to analogue converters) do exactly what they say on the tin. An ADC takes an analogue signal and converts it into a digital signal that the Raspberry Pi can process. A DAC does the opposite by taking a digital signal that the Raspberry Pi has produced and converting it into an analogue one.

Using I^2C we can connect four ADCs to the output pins of the joystick and be able to read what the joystick is doing. We can then also connect four DACs to the joystick input pins on the G2C and output signals to the wheelchair. It is possible to buy four channel ADCs and DACs which means you only need one chip for each rather than four for each.

Power

So far, we know how to read the receive signals from the joystick and we know how we are going to send out our own with the Raspberry Pi, but all the ADCs, DACS, and the Pi itself all need power. Now the keen eyed may have noticed that the Pi, ADCs, and DACs all run off 5V and so does the joystick so we can just use that, and all is good, right? Unfortunately, like most things in the world, it is not that simple.

The GC2 derives its 5V power supply on board using the 25V input voltage and is only designed to power the components of the GC2. It is not designed to have an entire computer syphoning off some of this power. You might then go on to say, "well gee, why can't we do that then?" The 25V supply is easily enough to support the Pi and all the other components, all we need to do is convert the voltages which we can do with a buck converter. A buck converter is a type of switched-mode power supply that steps down a voltage. They are typically very efficient at about 95%. To power the Raspberry Pi, we need a buck converter that will accept 25V as an input and will output 5V. This will be enough to power the Pi, but we cannot use this to power the ADCs and DACs.

Most consumer ADCs and DACs use an internal voltage reference that is typically derived from their main power input. This reference voltage is used to figure out what voltage is coming in/should be going out of the chip. This means that the power source for the ADCs and DACs must be the same as whatever is on the other end, in this case the joystick. If the joystick, ADCs, and DACs are powered from different power supplies, then any difference between the power supplies will show as a change in the signal, even if the signal did not change. The only way to prevent this is by using the same power supply for the joystick, GC2, ADCs and DACs. The onboard 5V supply will be able to manage the addition of the ADCs and the DACs.

Unfortunately, it does not end there. If we were to connect the pi to the buck converter and then connect the ADCs and DACs to the joystick and then connect them together, nothing would happen. This is because I^2C also needs a constant reference voltage to work correctly. By having these two sets of components split across power supplies, they end up with different reference voltages so neither can communicate. Conveniently though, the reference for I^2C is 0V, not the positive power supply so all we have to do to solve this is join the 0V connection of the Raspberry Pi to the 0V connection of the wheelchair giving a common 0V across all the components.

So, to put all of that simply, the ADCs and DACs get powered by the GC2 so they can work with the joystick, the Raspberry Pi gets powered by the 25V input of the GC2 via a buck converter so it can have enough power to work, and the 0V line of the Raspberry Pi and the GC2 is joined so that everything can talk to each other.

Modification and Assembly

Now we know the theory behind controlling an electric wheelchair, we can put it into practice. The assembly phase will be done in two parts, modifying the GC2 and assembling the Raspberry Pi control unit.

The GC2 and the Raspberry Pi control unit are then connected using two RJ45 cables. As each RJ45 cable has eight cores inside, all the joystick signals, both send and return, can travel down one cable, and all the power can go down the other cable. By being clever with which signal goes down which wire inside an RJ45 cable, you can easily bypass the Raspberry Pi controller using a third RJ45 connector to create a loopback connector.

Part 1: Modifying the GC2 joystick controller

Ensure you power off the wheelchair and disconnect the GC2 joystick controller unit from the control harness. Remove any cable ties holding the GC2 cable onto the wheelchair. Remove the GC2 from the wheelchair by removing the two crosshead screws on the base of the GC2.

Next, open the GC2 by using a hex key to remove the three small screws on the underside of the GC2. The cable is mounted inside on the bottom half of the GC2 and connects to the circuitry in the top half of the GC2 using a white connector. Make sure to disconnect this before you fully separate the halves to avoid breaking the internal wires.

Next, drill two holes roughly 5.2mm in diameter into the into the upper half of the GC2 housing to allow the RJ45 cables to pass into the GC2. Drilling on the top half gives you space for the cables to come in and stops them from interfering with the wheelchair armrest when it is installed again. Be aware of the supporting pillars in the bottom half of the GC2. Make sure the holes that are drilled do not interfere with these.

Optional (but advised) Emergency-stop button

If installing an emergency-stop button, drill a third hole for the emergency-stop cable. You may want to drill this in the bottom half of the GC2 to allow for easy assembly.

Next locate the red power cable from the cable harness and cut it between the ferrite coil and the charging port, allowing yourself enough cable to solder on to each side.

Pass the 2-core cable through the third hole and strip the ends. Place some heat-shrink tubing on each wire and then solder one wire to each side of the cut cable. Use the heat gun to heat-shrink the tubing covering the solder joint.

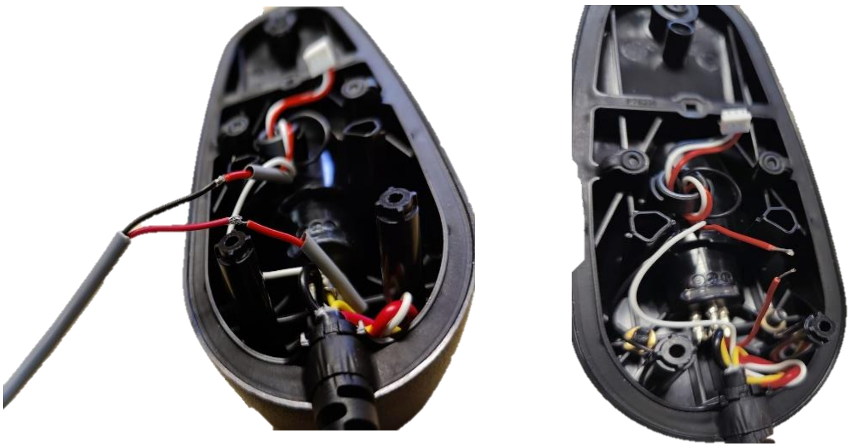


Figure 6 Emergency-stop cable modification

On the other side of the 2-core cable, connect the emergency-stop button to the normally closed (NC) connections. This will allow power to flow normally but if the button is pressed, power will be cut to both the to the GC2 and the Raspberry Pi. This will also stop the wheelchair completely.

Circuit board removal

Figure 7 The GC2 circuit board

Next, we need to remove the circuitry from the GC2 casing. Turn the controller around so you can see the circuit board. The speed control potentiometer needs to be removed from the casing before the circuit board can be removed. The potentiometer is mounted in a pillar and held in place with a small clip. This clip is on the pillar on the bottom side of the potentiometer near the solder joints for the wires. Push this clip back in the direction of the joystick to remove the potentiometer. It should slide out easily. If you face significant resistance, stop. There should be enough length in the three cables connected to it to allow it to come out easily.

The circuit board is now held in place with three prongs, two at the bottom near the buzzer and one on the pillar that held the potentiometer. Slide the circuit board in the direction of the joystick. This requires a small amount of force. Again, if you face significant resistance, stop.

Remove the joystick cap on the outside of the GC2 enclosure to reveal the metal rod beneath it. The black ribbed dust cover (the pyramid shaped bit) does not need to be removed. The joystick is now held in by two clips from the inside of the enclosure. These two clips are on the left and right of the joystick. Both need to be pushed back to release the joystick, but it is possible to do one at a time.

Joystick signal wiring

Now that the circuit board and joystick are free from the enclosure, locate the ribbon cable going to the joystick and without cutting them, separate each of the individual wires using a craft knife so that you can access each wire in the cable easily. Be careful not to damage any of the wire insulation. See Figure 8 for what we mean by separate.

We will work with the joystick at the top of our circuit board and the speed control at the bottom. The missing side of the power connector on the circuit board should be at the top of the board.

On the side of the circuit board that has the black power connector, and going from right to left, cut wire 2 (Left/Right 1), wire 4 (Forward/Backward 1), wire 5 (Forward/Backward 2), and wire 7 (Left/Right 2) allowing yourself enough wire to solder on to each side. Strip each wire enough to solder on to. You should now have eight exposed wires, four on the joystick side and four on the circuit board side.

Next, cut one connector from one of your ethernet cables and pass it through one of the first two holes in the controller. Ethernet cables use two wiring standards so make sure each wire in the cable is in the same order on both ends of the cable! This guide used the T568B standard. Remove the outer insulation, separate out each wire and strip the ends enough to solder on to.

Add heat-shrink tubing to each wire before soldering them together!

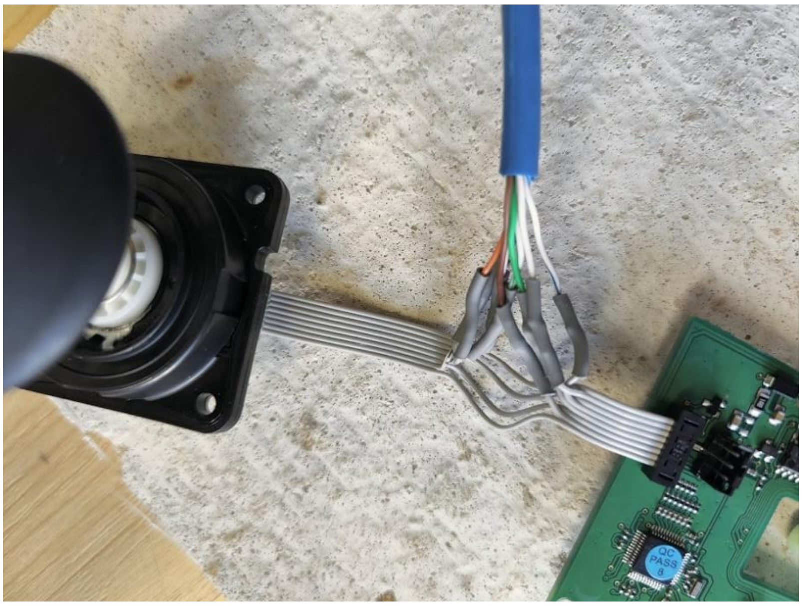


Figure 8 Joystick send and receive wiring with the wires seperated out

Working from right to left, on the joystick side of the ribbon cable, solder wire 2 (Left/Right 1 Send) to the green and white wire (RJ45 pin 3), wire 4 (Forward/Backward 1 Send) to the orange and white wire (RJ45 pin 1), wire 5 (Forward/Backward 2 Send) to the orange wire (RJ45 pin 2), and wire 7 (Left/Tight 2 Send) to the blue wire (RJ45 pin 4). Use the heat gun to heat-shrink the tubing covering the solder joints.

Next working from right to left on the circuit board side of the ribbon cable, solder wire 2 (Left/Right 1 Receive) to the brown and white wire (RJ45 pin 7), wire 4 (Forward/Backward 1 Receive) to the blue and white wire (RJ45 pin 5), wire 5 (Forward/Backward 2 Receive) to the green wire (RJ45 pin 6), and wire 7 (Left/Right 2 Receive) to the brown wire (RJ45 pin 8). Use the heat gun to heat-shrink the tubing covering the solder joints.

You should now have one cable that will handle all the joystick signals. We have used these specific RJ45 pins to make the loopback connector easier to make.

Power wiring

This next ethernet cable will handle all the power connections. Ideally you want to use a distinct colour cable, so you do not get the cables confused.

Cut one connector from the other ethernet cable and pass it through the last hole in the controller. Remove the outer insulation enough to allow wires to reach both sides of the circuit board easily. Strip the ends of each wire but try to keep the pairs together. Be careful not to strip too much off these. The orange pair in this second cable can be used as an optional switch that the Raspberry Pi can read from and could be used to change modes or to turn it on and off. We will not use it so feel free to leave these wires unstripped. All the soldering on this cable will be to pads on the circuit board so heat-shrink tubing is optional. If you have accidentally stripped the wires too much so that there is a risk that the components will short circuit, add some heat-shrink tubing to re-insulate it.

Solder together the two blue and blue and white wires (RJ45 pins 4 and 5) to form one wire. Next solder together the two brown and brown and white wires (RJ45 pins 7 and 8) to form a second wire. We are doubling up these cables as they are carrying a higher voltage. Sending power down these colour pairs matches the existing 802.3af PoE standard (this does not mean it is PoE compatible, we are just using the same colours).

Locate the three-pin power connector on the circuit board near to where the joystick cable is attached. The missing side of the connector should be facing the top of the board. There are three metal pins coming from this connector in a triangle shape. The top one closest to the ribbon cable connector is the signal cable. The pin on the bottom right is the 25V pin. The pin on the bottom left is the 0V pin.

Solder the blue and white pair of wires (RJ45 pins 4 and 5) to the 25V pin on the bottom right of that connector where it meets the circuit board. Be careful not to damage the other components around this connector or the connector itself as you do this. Make sure to leave enough clearance between the wire you solder on and the circuit board to avoid short circuits.

Next, solder the brown and white pair of wires (RJ45 pins 7 and 8) to the 0V pin on the bottom left of that connector where it meets the circuit board.

Once you have soldered the main power connectors, flip the circuit board around so you can see the solder joints for the ribbon cable connector. The black power connector should be on the other side of the circuit board now and the joystick should still be on the top. The top left solder joint of the ribbon cable is pin 1, pin 2 is beneath it, pin 3 is to the right of pin 1 and pin 8 is the bottom right solder joint.

Carefully solder the green and white wire (RJ45 pin 3) to the pin 1 solder pad (Joystick 5V) being careful not to accidentally bridge any of the solder pads together.

Next carefully solder the green wire (RJ45 pin 6) to the pin 3 solder pad (Joystick 0V) just to the right of pin 1.

Now check all your wires are soldered to the correct places. An incorrect wire can irreparably damage all your components, the wheelchair, or both (ask me how I know).

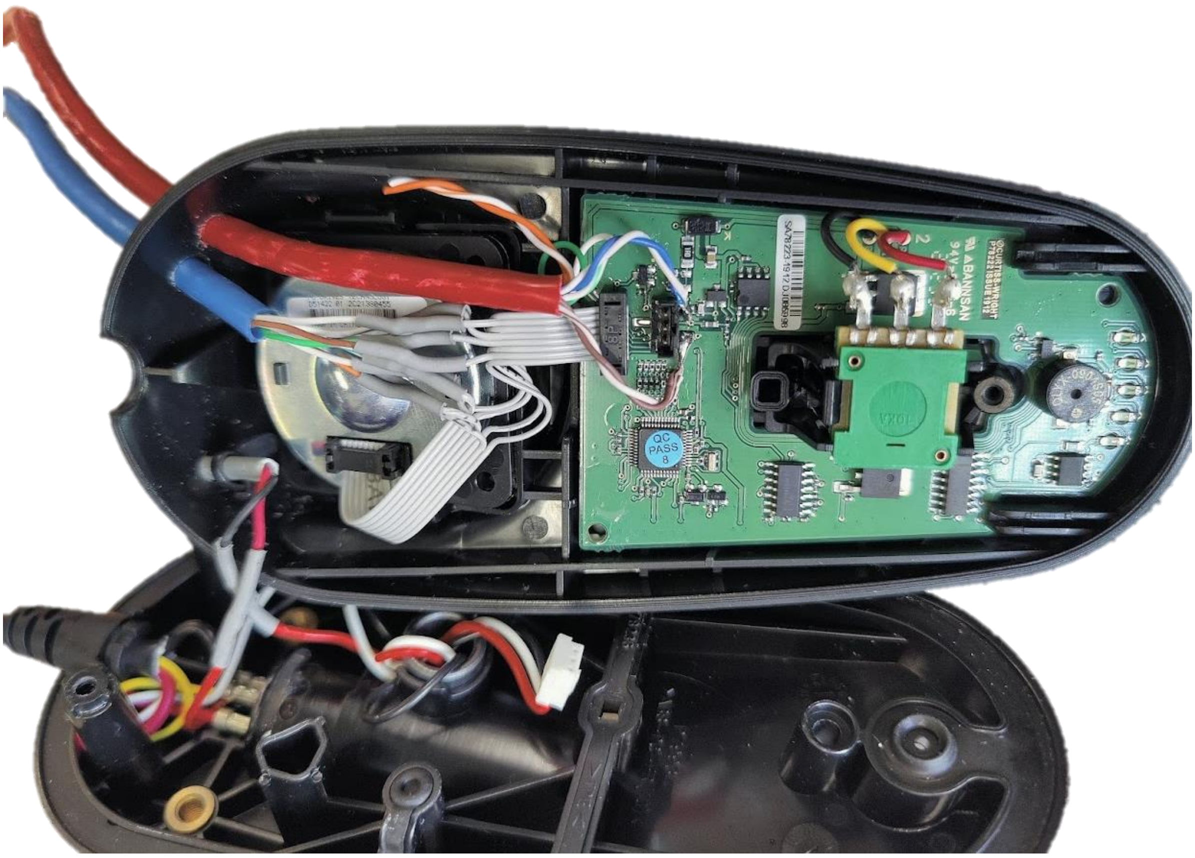


Figure 9 Completed internal wiring
Red cable is power, blue is joystick signalling, and grey is for the emergency stop

Reassembly

Now you have checked all your wiring and you know everything is correct, the modified GC2 can be reassembled. Re-attach the circuit board into the top half of the GC2 enclosure and re-mount the joystick. Note, the joystick is keyed to only fit in one way. Do not force the joystick back in. The joystick cap can now be put back into place. Re-mount the speed control potentiometer. If it is not going in correctly, check the flat wedge on the moving bit of the potentiometer aligns with the flat bit on the inside of the dial. Thin cable ties can be added to the two ethernet cables and the optional emergency-stop cable on the inside of the GC2. These act as strain relief to prevent damage to the solder joints.

Finally reconnect the power connector from the bottom half of the GC2 to the circuit board. Make sure the ferrite ring is secure in the bottom half of the GC2. It should slot into a circular mount on the inside of the bottom half of the GC2. You can now close the GC2. The GC2 has several support pillars in the bottom half to support the circuit board, support the joystick, and to add structural rigidity. Make sure the new cables do not interfere with these as you close the GC2.

Loopback connector

The keen eyed among you may have noticed something interesting about the ethernet wiring for the joystick. All the joystick sends are on one side, and all the joystick returns are on the other. By connecting the ethernet cable in this way, we can easily create a loopback connector. This will allow the joystick to control the wheelchair like nothing had ever happened, reversing the modification whilst it is connected. This is useful if you want to remove the Raspberry Pi and still use the wheelchair.

Figure 10 Loopback connector

For any researchers or academics, connecting the loopback allows for an easy comparison between different control systems and a controlled test without a control system.

To make the loopback connector, make four short wires and strip both ends. Take one of the RJ45 breakouts and solder each wire to the first four pins.

Now solder these wires back into the last four pins (See Figure 10 for reference). This should connect pins 1 and 5 (Forward/Backward 1), 2 and 6 (Forward/Backward 2), 3 and 7 (Left/Right 1), and 4 and 8 (Left/Right 2).

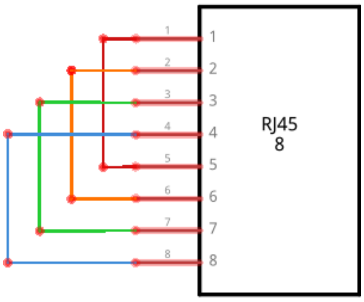


Figure 11 Loopback connector schematic

Test your loopback connector to ensure the pins are correct before using it. Mismatched pins will not damage the wheelchair, but it may result in unexpected behaviour such as swapped directions or the GC2 temporarily locking out.

Part 2: The Raspberry Pi controller

Your circuit should be assembled according to this circuit diagram (Figure 12 Circuit diagram for the Raspberry Pi controller). We strongly recommend using a protoboard with a similar layout to a breadboard for easy use.

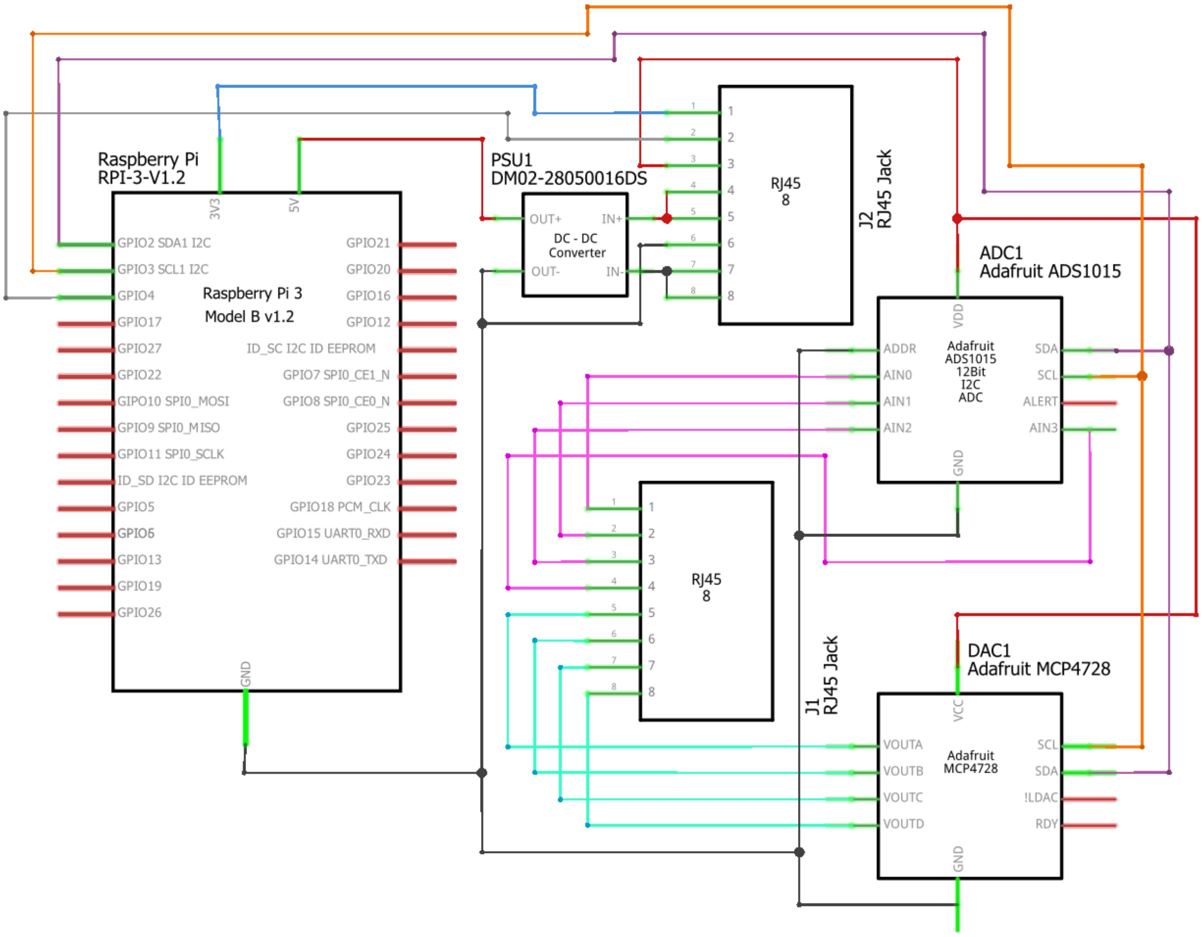


Figure 12 Circuit diagram for the Raspberry Pi controller.
In this diagram, red is V+, black is 0V, orange and purple are the I²C connections, blue and grey are the optional switch connections, magenta is the joystick send wires, and aqua is the joystick receive wires.

Protoboard circuitry

First solder the headers onto the ADC, DAC, and buck converter boards. To keep the headers straight, you can use a breadboard to hold the headers whilst you solder them.

Next, solder the female HAT header onto the protoboard. This will connect to the Raspberry Pi so make sure this is mounted on the bottom of the protoboard.

Once all the headers are in place, solder the ADC, DAC, and buck converter boards onto the protoboard. Make sure that each pin on each board is connected to its own track on the protoboard to prevent damaging your components or wheelchair.

Once this is done, wire the rail connected to the positive output of the buck converter to one of the 5V pins on the Raspberry Pi header on the protoboard, and the rail connected to the negative output of the buck converter to any of the 0V pins on the Raspberry Pi header. This will power the Raspberry Pi using the 25V source on the GC2.

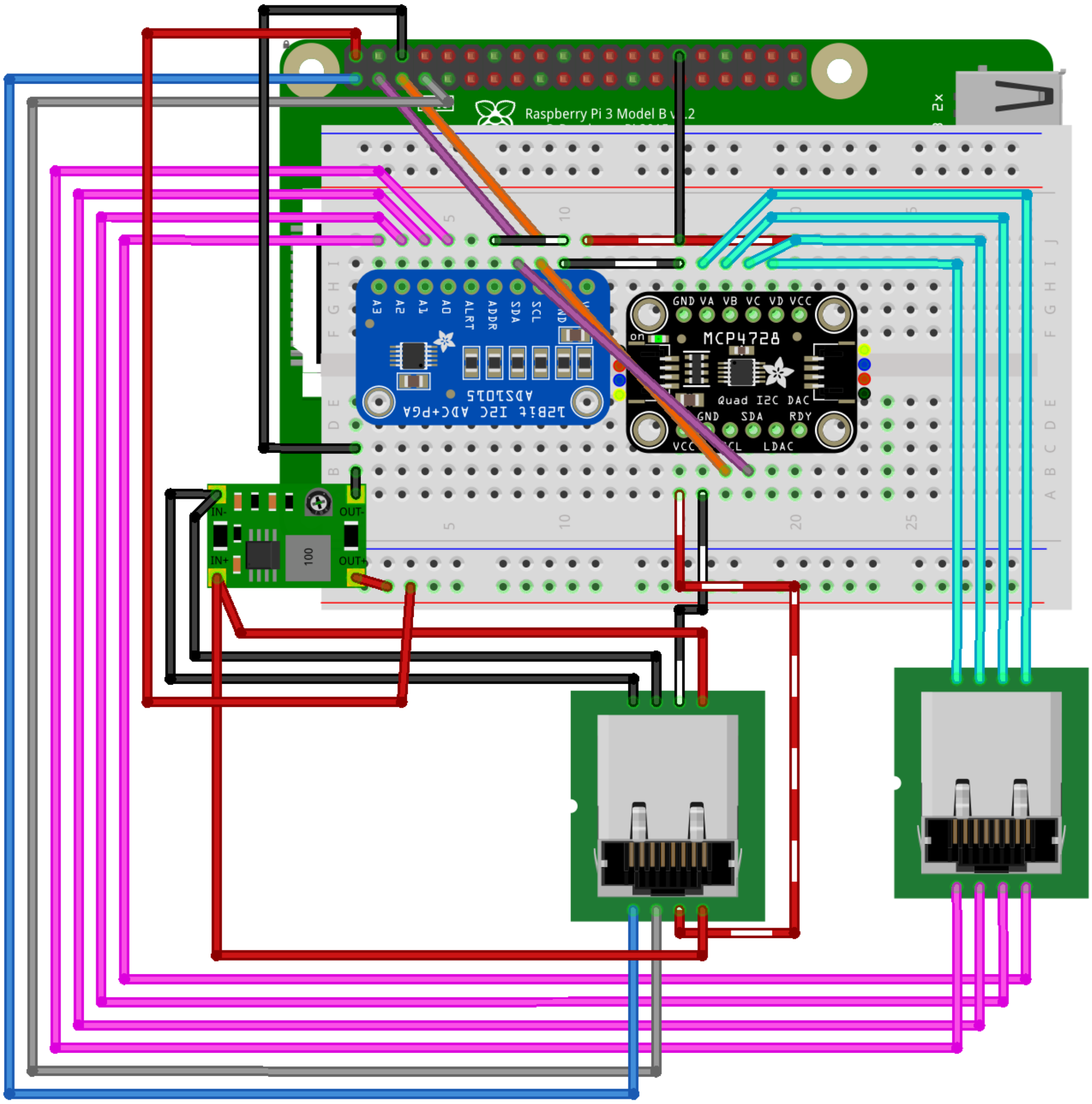


Figure 13 Protoboard component layout

Next connect the rails connected to the V+ pins of the ADC and DAC together (these are normally labelled VCC) and then do the same for the 0V pins (normally labelled GND). Once this is done, connect one of the 0V rails on either the ADC or DAC to a 0V pin on the Raspberry pi header. This is our common ground connection so the Raspberry Pi, ADC, and DAC can all use I^2C.

Now connect the SDA pins on the DAC and ADC together and then connect these to the SDA pin on the Raspberry Pi header. Then do the same for the SCL pins. The SDA and SCL connections are the I^2C communication connections.

RJ45 breakouts

Take each of the RJ45 breakout boards and solder a wire to each pin. One breakout will be used for joystick signals and the other for power. We strongly recommend that you label each breakout to avoid confusion later. Miswiring these connectors can irreparably damage all your components, the wheelchair, or both.

Joystick breakout

Take the breakout connector that you wish to use for the joystick signal and identify pin 1 on it. If pin 1 is not labelled, you can plug in an ethernet cable and pin 1 will be closest to the orange and white wire and furthest away from the brown wire. Connect the wires from pins 1, 2, 3, and 4 and connect them to the analogue inputs of the ADC. These are normally labelled A0, A1, A2, and A3.

Next, connect the wires from pins 5, 6, 7, and 8 and connect them to the analogue outputs of the DAC. These are normally labelled as V0, V1, V2, and V3 or VA, VB, VC, and VD. Be careful not to mistake VC and VCC if this labelling scheme is used on the DAC.

Power breakout

Take the remaining breakout connector that you wish to use for the power connections and identify pin 1 on it.

Connect pin 3 to the V+ pins on either the ADC or the DAC. This is the power supply and reference voltage for the ADC and DAC. Connect pin 6 to the 0V pin on the Raspberry Pi header, the ADC, or the DAC. The 0V rail is commoned across all the circuit so it does not matter which 0V connection you use.

Next connect pins 4 and 5 to the positive input of the buck converter and pins 7 and 8 to the negative input. This is the 25V power supply that the buck converter will convert to 5V for the Raspberry Pi to use.

Optionally if you wanted to use a switch on the orange pair that we left in the GC2, connect pin 1 to the 3.3V pin on the Raspberry Pi header and connect pin 2 to one of the GPIO pins (we recommend GPIO4). When the switch is on, the Pi will see 3.3V on the GPIO pin, and when the switch is off, the Pi will see 0V on the GPIO pin.

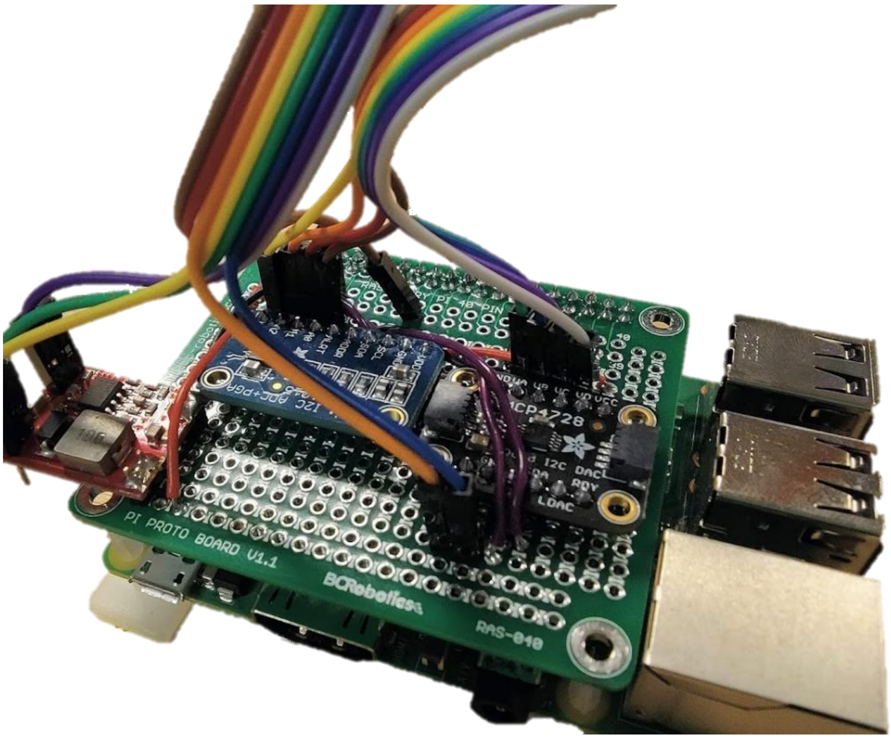


Figure 14 The fully wired protoboard mounted on the Raspberry Pi

Mounting the circuitry

Mark out where your RJ45 and USB passthrough connectors will go on the enclosure and then drill the holes out for them. Ideally you want to position the connectors on the sides of the enclosure, so you do not get any water coming into the ports from the top or splashed up from the bottom. Mount the connectors onto the enclosure ensuring a tight fit so they do not spin or let water in. Rubber washers can be optionally used to increase watertightness.

Once the connectors are mounted, label the RJ45 connectors to avoid confusion. Connecting the wrong cable to the wrong socket can irreparably damage all your components, the wheelchair, or both (also unfortunately learnt from experience).

Next, attach an adhesive PCB standoff into each of the mounting holes in the Raspberry Pi. The Raspberry Pi mounting holes may need to be expanded to fit the standoffs correctly. These mounting holes can be significantly expanded before running into any issues.

Trim any component legs from the bottom of the protoboard using a pair of side cutters and mount the protoboard onto the Raspberry Pi. Make sure pin 1 of the protoboard matches pin 1 of the Raspberry Pi.

Connect the USB panel connectors to the USB sockets on the Raspberry Pi and then connect the RJ45 breakout boards to the RJ45 panel connectors being incredibly careful to connect the breakouts to the correct panel sockets.

Finally remove the covers from the adhesive feet and stick the Raspberry Pi into the enclosure.

Figure 15 Completed Raspberry Pi controller fitted in the controller.
Notice the two different colours of ethernet cables used.

Mounting the controller on the wheelchair

The GC2 can now be mounted back onto the armrest and the Raspberry Pi enclosure can be attached to the underside or rear of the wheelchair. Make sure that any cabling going to the controller sags beneath the connection ports so that any water that runs along the cables will drop off rather than running into the connectors.

When mounting the optional emergency-stop button, be sure to mount it in a place that is easily reachable but still takes some effort to activate. This is because you want to be able to activate it reflexively, if necessary, but you do not want the e-stop to become the go-to instead of the emergency option.

And that is it! You now have a fully remote controllable, programmable wheelchair!

Control the wheelchair

So now you have your shiny, new, and hopefully still working, modified electric wheelchair. The next question however is what you do with it!

The ADCs and DACs that were recommended both have easily accessible python libraries available from the manufacturer's website. Using these libraries and the online guides, you can receive signals from the joystick with a 12-bit integer value for each of the joystick channels. Forward/backward 1 will be on input 1, forward/backward 2 will be on input 2, left/right 1 will be in input 3 and left/right 2 will be on input 4. As we don't have access to the centre tap wire on the joystick, we need to estimate where the middle value of the joystick is. The simplest way to do this is to leave the joystick still for a while and see what value the joystick outputs on each channel. If the project is working correctly, they should all be roughly the same value.

When you move the joystick forward, you will notice that the values forward/backward 1 and 2 will both move together. By measuring the difference between these, and adding a threshold, you will be able to tell if the joystick behaves in a faulty way in the future. This is what the GC2 does normally.

When outputting to the wheelchair using the DAC, you must set all the output channels to have a non-zero voltage otherwise the GC2 will complain of a joystick error. We recommend the middle value of the joystick so that it stays still. You set the voltage using a 12-bit integer, similar to the values you get from the ADC. To make the wheelchair move, just increase, or decrease this number across the first two channels for forward/back and the last two channels for left/right. Remember, the values for channels 1 and 2 for each direction must not have a large difference between them otherwise the GC2 will assume there is a fault and lock-out the controller.

Table of Figures

References

[1] Pride Mobility Products Corp, “Jazzy Select 6,” 10 2021. [Online]. Available: https://pride-mobility.co.uk/wp-content/uploads/2021/10/UK_Select_Series_om_RevF_Feb12_3244.pdf. [Accessed 29 04 2022].

[2] Curtiss-Wright, “JC2000 Multi-axis contactless joystick controller,” 2017. [Online]. Available: https://www.cw-industrialgroup.com/getattachment/a6bb9db0-f4be-47c9-8e9d-e8931694969f/jc2000_brochure. [Accessed 04 29 2022].