SV MCU V2

Writer: Ben Hyslop

Written: 05/02/2026 (last night)

Times Higher Education University of the Year 2012 & 2019
Times Higher Education Widening Participation Initiative of the Year 2019
The University of Strathclyde is rated a QS 5-star institution

Contents

0 Abstract

This project is the 2nd version of mcu designs made. It solves some of the inexperienced problems with V1 whilst also adding a bit more functionality with some additional pins. The main objective of this project was to design a faster and more affordable board than the Arduino Uno using one of the STM32 chips and allow for a more efficient learning experience with microcontrollers for new members to the society. Specifically the chip is the STM32G474RET6. While the board is faster than the Arduino by 150MHz it is more expensive by roughly £70 due to shipping and assembly costs. However to lighten the blow on the expenses this does open up an exciting venture for learning that many other societies cant offer. As well the fact it works proposes that many other parts can be made and printed onto another PCB or rather be more modular. This piece does not plan on discussing how the code works . That will be a sperate document yet to be written. This only entails the hardware and pcb designing of the board. It will not delv specifically aswell on how to use software for pcb design as there are very slight changes from software to software and its within the best interest of everyone not to be stuck to the same software as even writing this there are considerations for using another software now. If considering just watch a tutorial on how to start with chosen software , play around with it and then go ham on whatever it is desired to be made. This document is only to explain the current design, its costs and provide a good idea of how to use it.

1 Introduction

First, it should be defined what a microcontroller actually is, what it needs and then what the project needs. A microcontroller is a small chip with everything needed in there for most projects. However, things like the clocks can be slightly inaccurate. Not really desirable. So the entire idea behind making this is to use other peripherals (Important components) that are very accurate, and then build around the main chip and then add in other funny stuff. Now it was mentioned that peripherals are important components. However it will be assumed that's not common knowledge. So what it actually is, is the group of components that will essentially support the cpu which is within the mcu. As mentioned some of its hardware can be slightly inaccurate due to the size limitations. This consists of things like clocks, timers, specific communication pins and adc's. Theyre are a whole lot more but not really, in need of being 'replaced' essentially. Clocks are the only things that need to be considered to be replaced as pins are well pins and the system currently runs with only PWM signals as far as im aware. Well maybe a comparator for diagnostics but nothing else really. So essentally all that needs to be added is a clock, preferably quite fast, additional memory and preferably speed controllers on the board itself. As well some pins are gonna be needed for the coding interface. Essentially how to programme the board but also how it can be debugged. Ie looking at i/o of pins. Hence why a compartor wont be needed on the board.

2 Background

The main contents of this chapter are brief summaries of what was written after researching each part of the MCU. These are merely to explain what is being seen.

2.1 Clocks

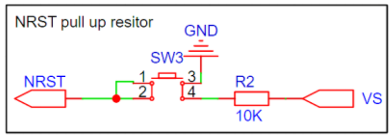
So a great thing about the stm32 mcu is it allows for external clocks. Limits of course depending on the mcu but the one being used only allows 4-48MHz. It was decided that a 16MHz clock would be used just for simplicity within the clock configuration ide. This isn't stuck to 16MHz. The mcu can still run at 170MHz it will only amplify the 16MHz signal. A crystal oscillator is incredibly useful. Basically it's a small crystal been tapped at a certain frequency. When tapped it releases an electromagnetic field. This tapping can come with a high chance of noise which is why capacitors are added to stop any from getting out. Capacitors are also added to match the capacitance with the oscillator which can also dispel some noise. Just ensure C1 and C2 equate to CL. A general equation to follow if no schematic can be found below. As well this does of course lead into some technical issues which will be delved into later. The schematic comes from the datasheet of the osccilator so any issue with the clock should be directed to the datasheet. General good advice is to keep the clock as close to the board as possible. This is so there is almost no chance of delay from the clock source to the pin.

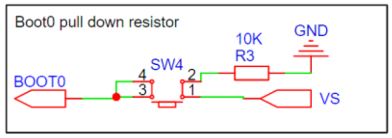
$$C_L = \frac{C_1 C_2}{C_1 + c_2} + C_{stray}$$

2.2 Reset Pins

So for programming purposes, we have two pins, NRST (which I didn't look up two days after setting up the pin) and boot0. Both pins reset the MCU, but play different parts. NRST is a negated reset state. So what was the board like initially. Peripherals, clocks, everything reset basically. Then boot0 is the pin that allows for reprogramming of the board. Both don't needs to be pressed in sequence just to rewrite the board. Only boot0 needs to be pressed, and NRST is just for safety measures currently. As can be seen in the pictures below, the pins are controlled by pull-up/down resistors. Basically, once the buttons are pressed what the signal without the resistor will take over for the signal that has the resistor in front of it.

If that makes any sense at all, but basically once NRSt is pulled down to ground by pressing the button, the whole system resets where as when BOOT0 is pulled up to VS it allows for code to be flashed onto the MCU.





2.3 SWD and JTAG

Turns out jtag is only useful with multiple mcu's or processors on the same board needing progarmmed. Much more simple to just use swd or serial wire data. This only uses 2 pins like i2c. SWDIO and SWCLK. SWDIO is our data transfer and SWCLK is the clock which presumably can run at 170MHz same as the board so very quick programming of the board.

Basically jtag uses atleast four pins; tck,tdi,tdo and tms. Theres a fifth known as a reset all the pins that have been affected from monitoring them. In a bit more detail it will ascynchornously reset the state of jtag operation. Essentially it will reset how the chip is being tested without affecting internal circuitry. Tdi and tdo are data in and outs. Tck is the clk and tms is the mode selected to be in when monitoring them, only issue is tms is based on lclk so the state will change only based on the clck timing. Also 2 bit synchronous traces can be added so to monitor whats coming in and out of the pins based on what’s been programmed. Fairly useful for debugging.

2.4 Memory shite

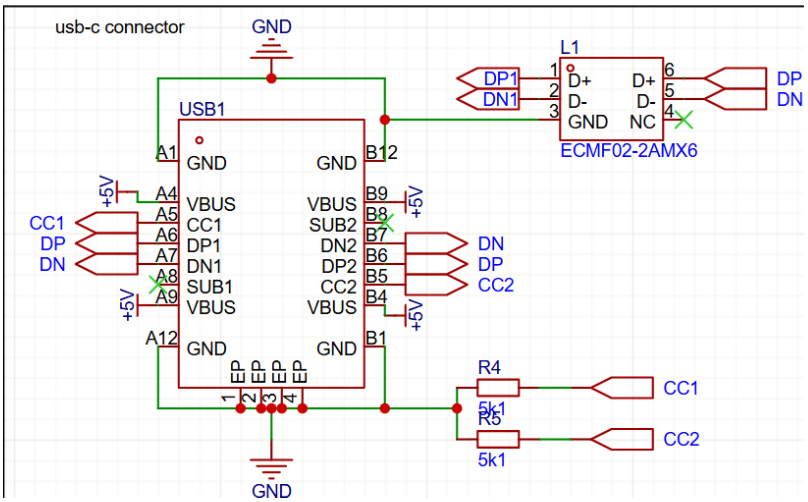
The memory chip being added only allows 1MHz. An inherent price for simplicity but the project isn’t at the level for adding RAM or trying to delve into communication protocols that require more than 2 wires. However extra memory is needed for safety. It's an I2C communication protocol that uses only 2 pins. SDA and scl. SDA is our datat transfer pin and scl is our clock.

2.5 3V3 regulator

Currently the board uses a LDO 5v to 3v downcoverter circuit found online. A very basic regulator but saves cost and space. Could have used a buckconverter which is found to be more conventional however for the sake of this project it’s a bit easier. There is 5V from the swd and usb connectors and that is how it should be preferred. Even if the pwr pins are screwed in, it would of the preference of everybody that no chance is taken. Also theres no need when the usb is how the jetson is already going to be communicating with the mcu.

2.6 USB-C Connector

One of the good additions of the V2 board is the usb-c connector. I wont delve to highly into how the communication protocol works as its not entirely necessary however what does to be know is that dp1 and dp2 are the same basically as well as dn1 and dn2. They are connected through the same calling as shown below. These pins are the main communication pins and because usb-c is a non rotary dependant port it was needed to make sure that either way it was turned the correct data was going through the correct port essentially. These pins are going through whats known as a vts diode which ensure protection against any lags from the usb (transients if you want to look it up). Finally, the 5V pins come through the USB connection inherently, and it is best practice to use them to their full advantage.



2.7 Schmitt trigger

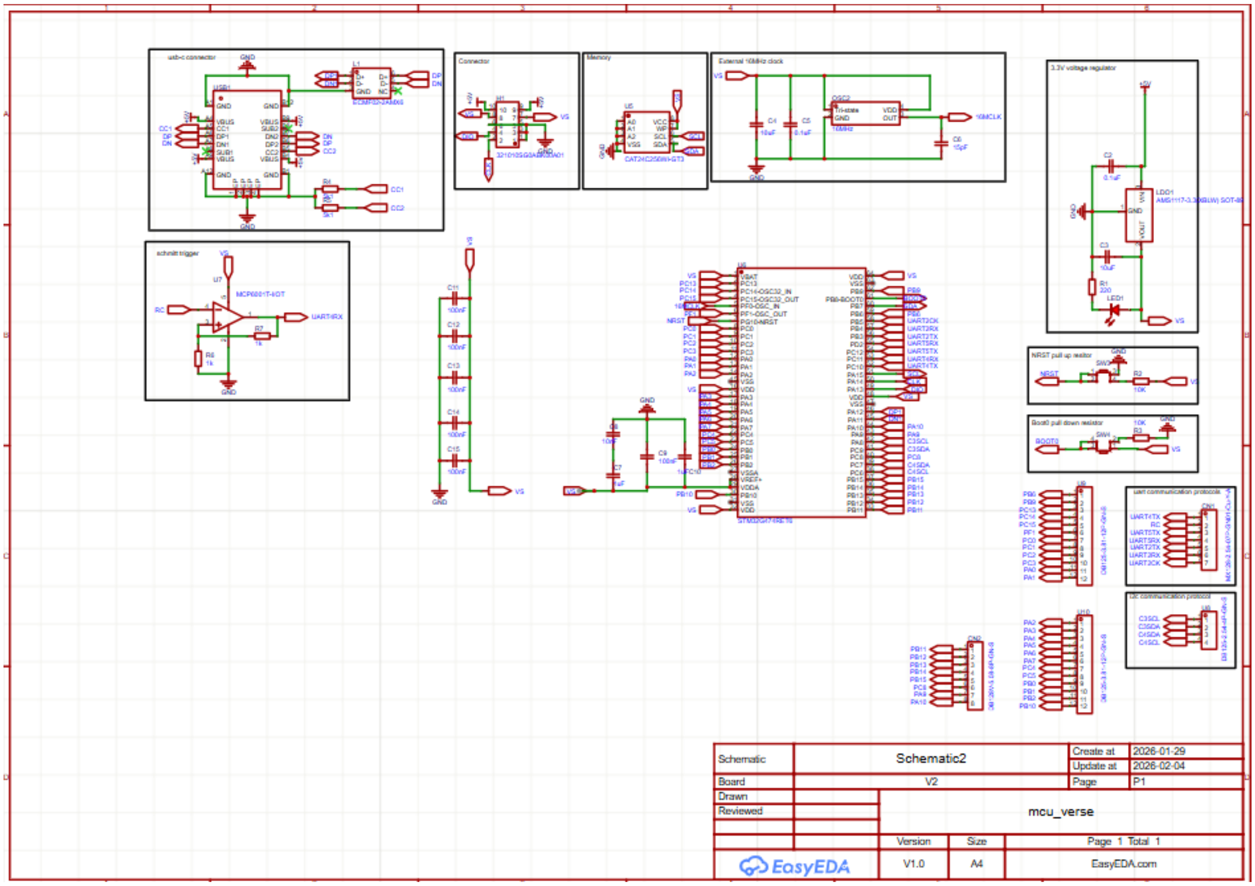
The Schmitt trigger is essentially a digital filter that ensures we get the correct data from even the noisiest of signals. This variation ive added is a inverting Schmitt trigger. Just because the data from the s-bus is an inverted uart (inverted binary signal) communication. And its better to not try fix this with code so was better to add this via hardware.

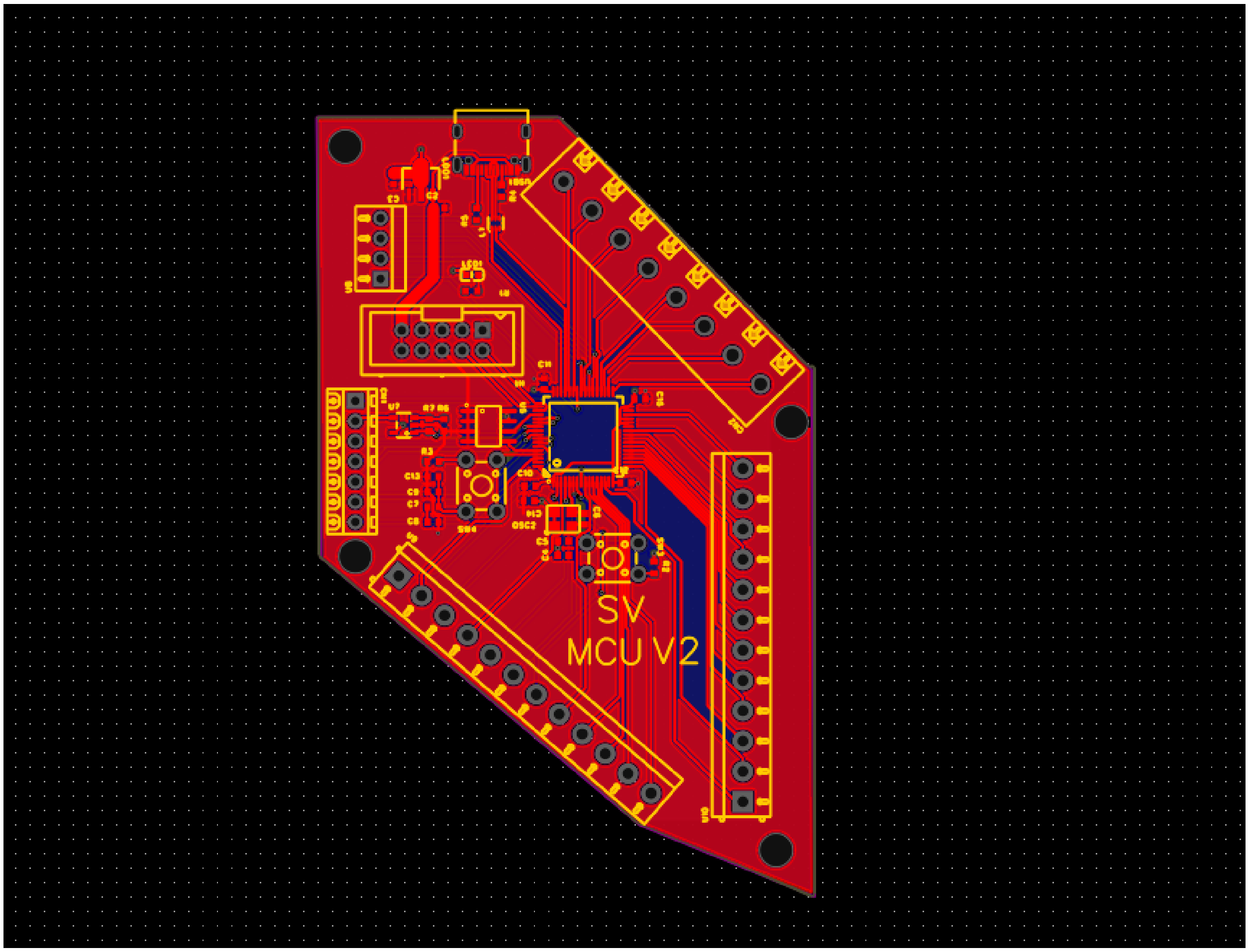
3 Issues and Drawbacks

Currently the board only appears to have minor issues with traces being either too close or not thick enough. This however is only solveable with more experience with pcb design but it should not have too great of an effect on the system as a whole.

Please just ensure that every component desired is there and if components are changed take enough time to change board layout and tracing. As well use as much of the board as possible ie some traces could be thicker. As well also look into other ideas for handling ghost current and voltages at the power supplies.

4 Schematics and layout of the board





The only thing concerning is the 2mm diameter or M1 screw holes on the board. Wether that's ok or not is fine they are easily changeable.

7 Datasheets

[1]- https://item.szlcsc.com/datasheet/CAT24C256WI-GT3/81132.html (memory chip)

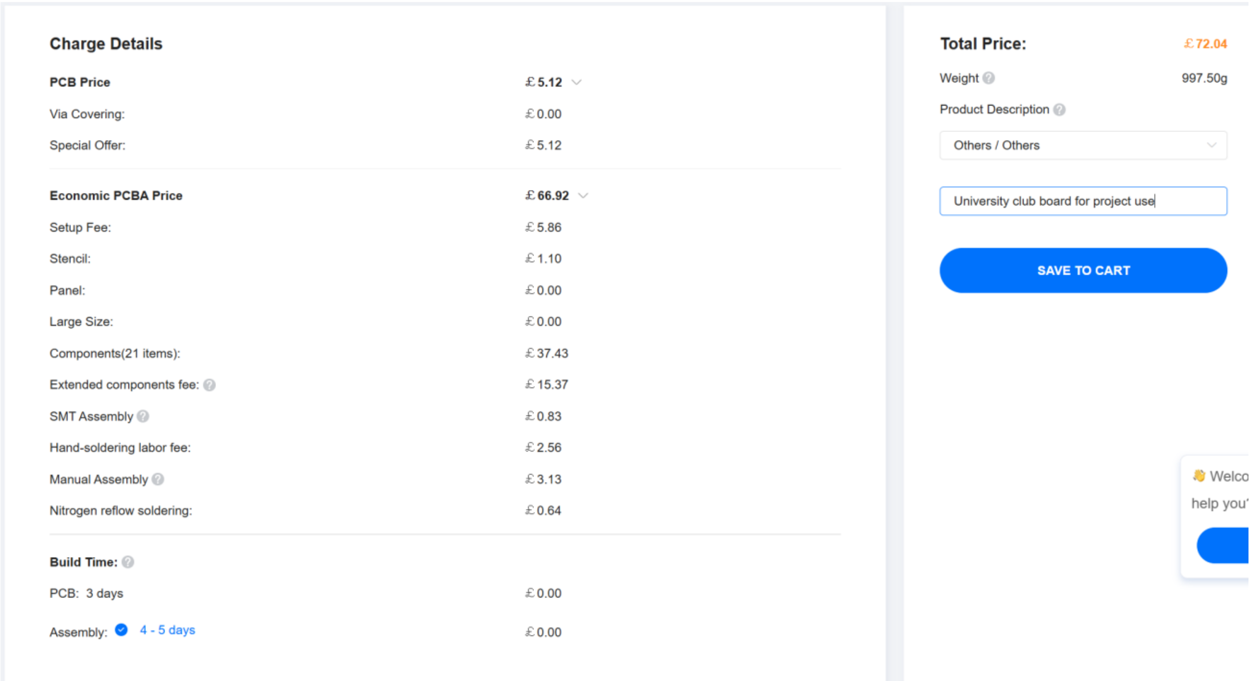
[2] - https://item.szlcsc.com/datasheet/OT2EL4C4JI-111OLP-16M/5867692.html (clock)

[3] - https://atta.szlcsc.com/upload/public/pdf/source/20181022/C283467_00098992BB1D287195B66486CE33A2DD.pdf (MOSFET)

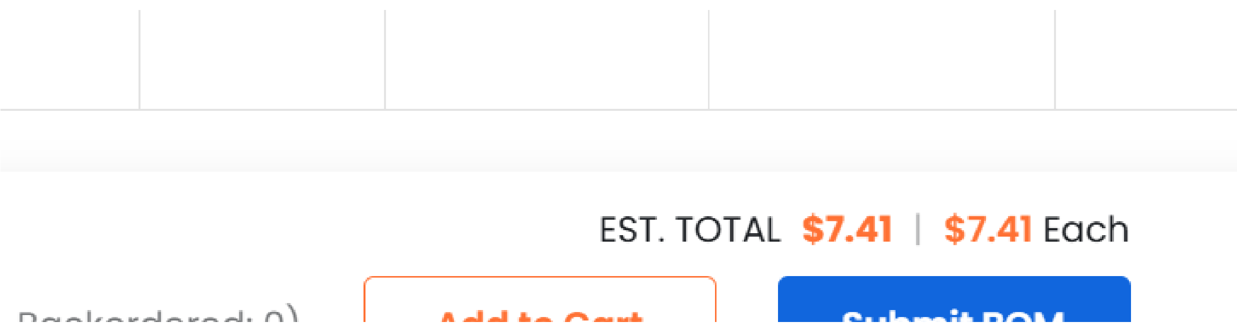
[4]- https://www.st.com/resource/en/datasheet/stm32g474cb.pdf (MCU)

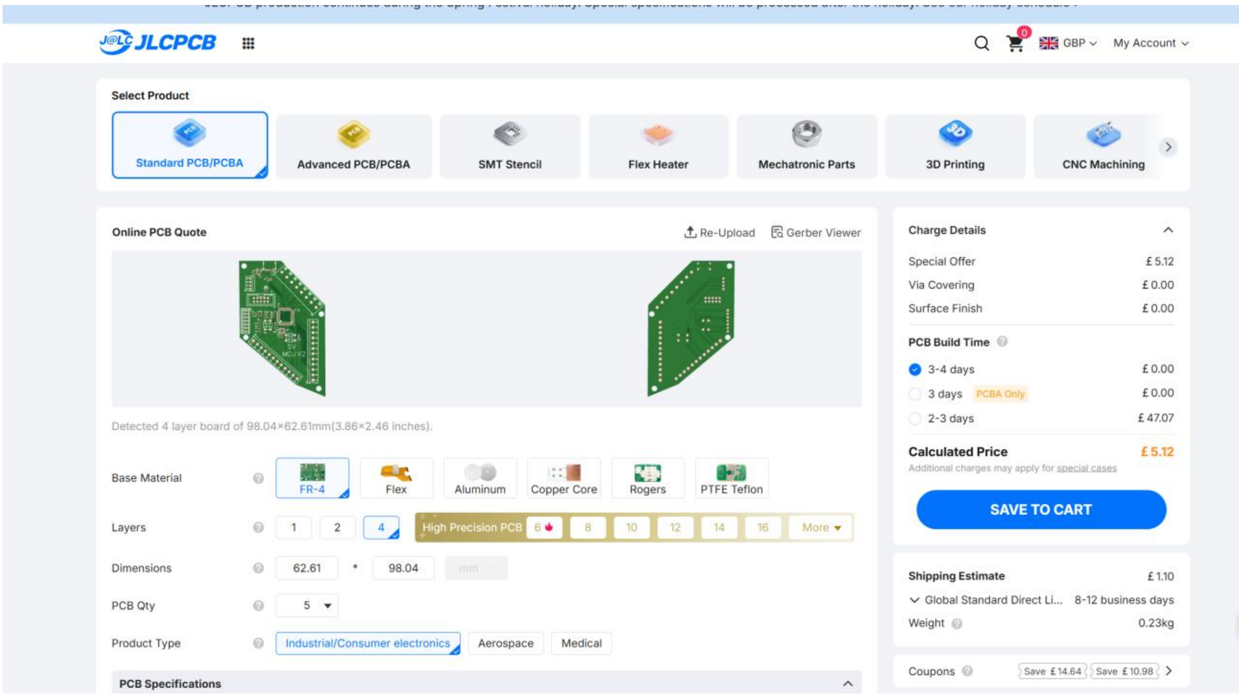
8 Cost breakdown

Now the total cost of getting the board made and assembled is roughly £72. The manufactures don't have usb-c connectors or switches that fit what I already have but its really no issue to solder myself.



The only other option is to have the board just built and sent to us for £6 and then component shipped separately for £8-9 depending on where it gets shipped to. Up to society but these are the set costs for the board.




JLCPCB
GBP
My Account
Select Product
Standard PCB/PCBA
Advanced PCB/PCBA
SMT Stencil
Flex Heater
Mechatronic Parts
3D Printing
CNC Machining
Online PCB Quote
Re-Upload
Gerber Viewer
Detected 4 layer board of 98.04×62.61mm(3.86×2.46 inches).
Base Material
FR-4
Flex
Aluminum
Copper Core
Rogers
PTFE Teflon
Layers
1
2
4
High Precision PCB
6
8
10
12
14
16
More
Dimensions
62.61
98.04
PCB Qty
5
Product Type
Industrial/Consumer electronics
Aerospace
Medical
PCB Specifications
Charge Details
Special Offer £5.12
Via Covering £0.00
Surface Finish £0.00
PCB Build Time
3-4 days £0.00
3 days PCBA Only £0.00
2-3 days £47.07
Calculated Price £5.12
Additional charges may apply for special cases
SAVE TO CART
Shipping Estimate £1.10
Global Standard Direct Li... 8-12 business days
Weight 0.23kg
Coupons
Save £14.64
Save £10.98