UNIPOLAR STEPPER-MOTOR DRIVER KIT. DATASHEET, BUILD INSTRUCTIONS, CIRCUIT EXPLANATION & CODE.

Overview: This kit once assembled creates a credit card sized circuit board that acts as a versatile driver for all shapes and sizes of unipolar stepper-motors. The device is designed to achieve very accurate control of the motors drive shaft. Making it an ideal driver for small and medium sized, low power robotics projects, where the precision and timing of a specific motion are crucial.

General Power Ratings & Usage:

Parameter	Min	Max	Units
Operating Voltage	5	18	**V**
Continuous Current	-	10	**A**
Operating Temperature	-10	100	**℃**
Output BEC Voltage	4.8	5.1	**V**
Output BEC Current	-	200	**mA**

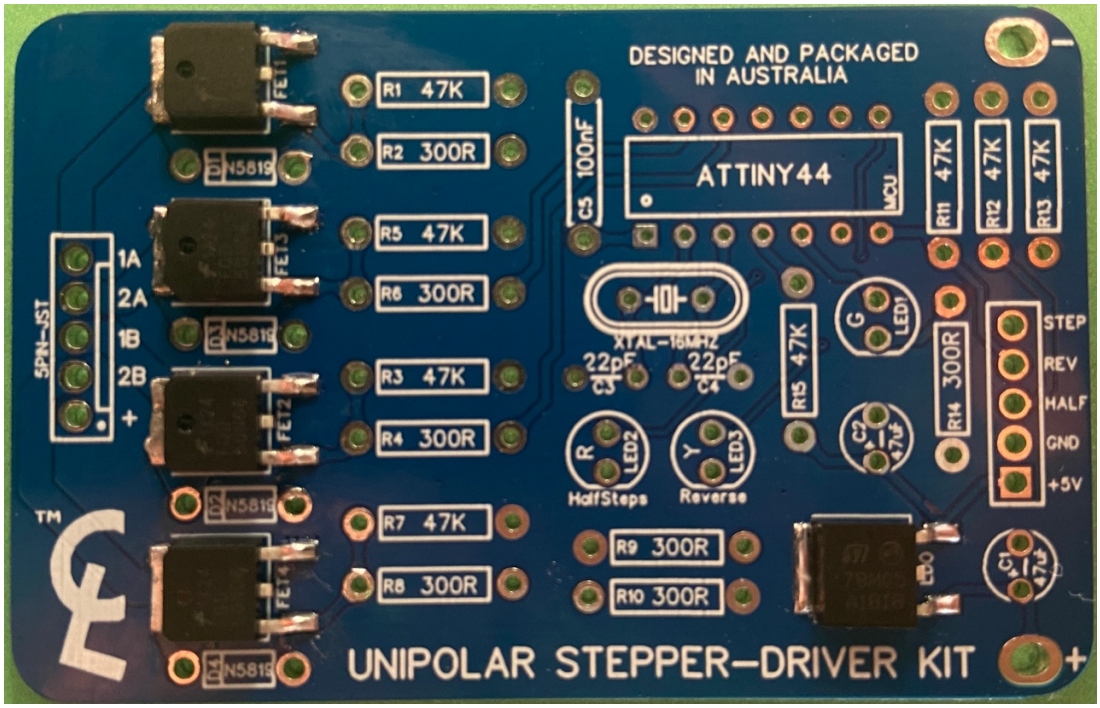
- To reach and maintain the maximum continuous current, heat sinks are required for all MOSFETS.
- The BEC (battery eliminator circuit) also supplies 5V power to the board meaning excess current draw will impact the driver's performance.
- A high pulse (5V) on the step input pin will cause the driver to take 1 step. Pulling the pins labelled rev & half logical high (5V) will trigger the reverse and/or half step drive modes.
- **Touching the board during operation, especially the power stage is not recommended**

Component List/BOM:

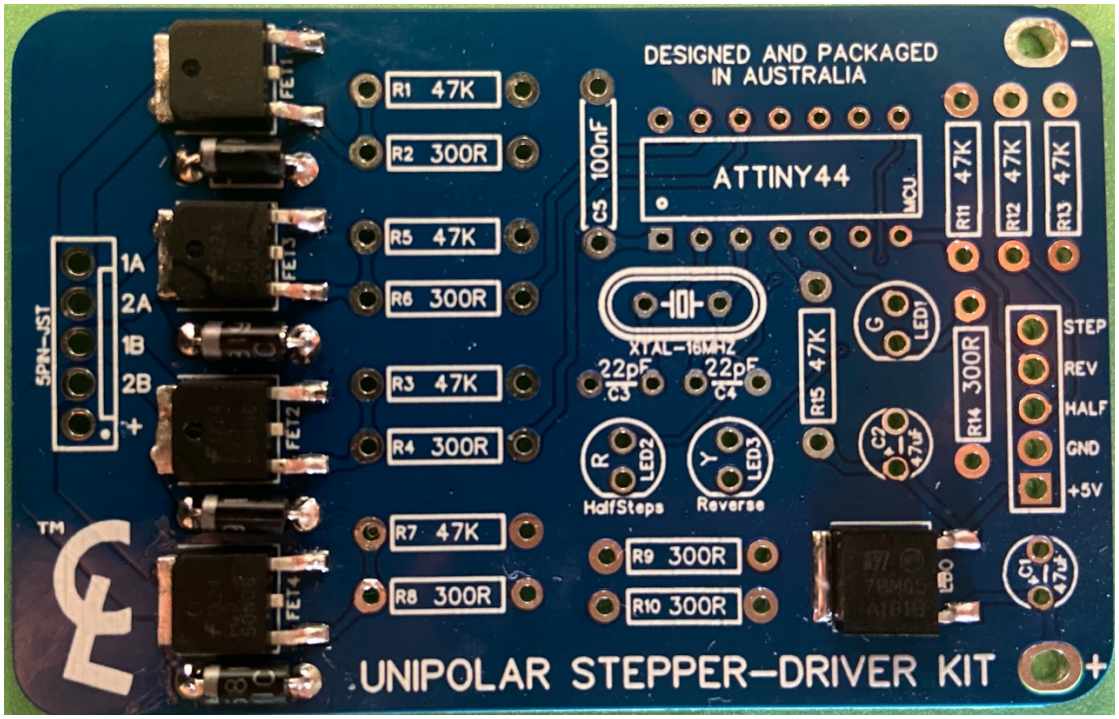
Quantity	Name	Type	PCB Designation
1x	PCB	Unipolar Stepper kit printed circuit board	-
4x	FQD50N06	N-Channel MOSFET	FET1 - 4
4x	1N5918	1A Schottky diode	D1 - 4
1x	5 pin JST connector	Motor JST Connector	5PIN-JST
1x	L78M05	5V linear regulator	LDO
1x	Attiny44	Microcontroller	MCU
1x	5 row of header pins	Header pins	STEP REV HALF GND 5V
3x	R, G, Y.	Colour LEDs	LED1 - 3
1x	XTAL	16Mhz crystal oscillator	XTAL-16MHZ
2x	22pF	Ceramic capacitor	C3, C4
2x	47uF	Electrolytic capacitor	C1, C2
1x	100nF	Film capacitor	C5
8x	47K	0.25W 1% metal film resistor	R1, R3, R5, R7, R11, R12, R13, R15
7x	300R	0.25W 1% metal film resistor	R2, R4, R6, R8, R9, R10, R14

Assembly Instructions

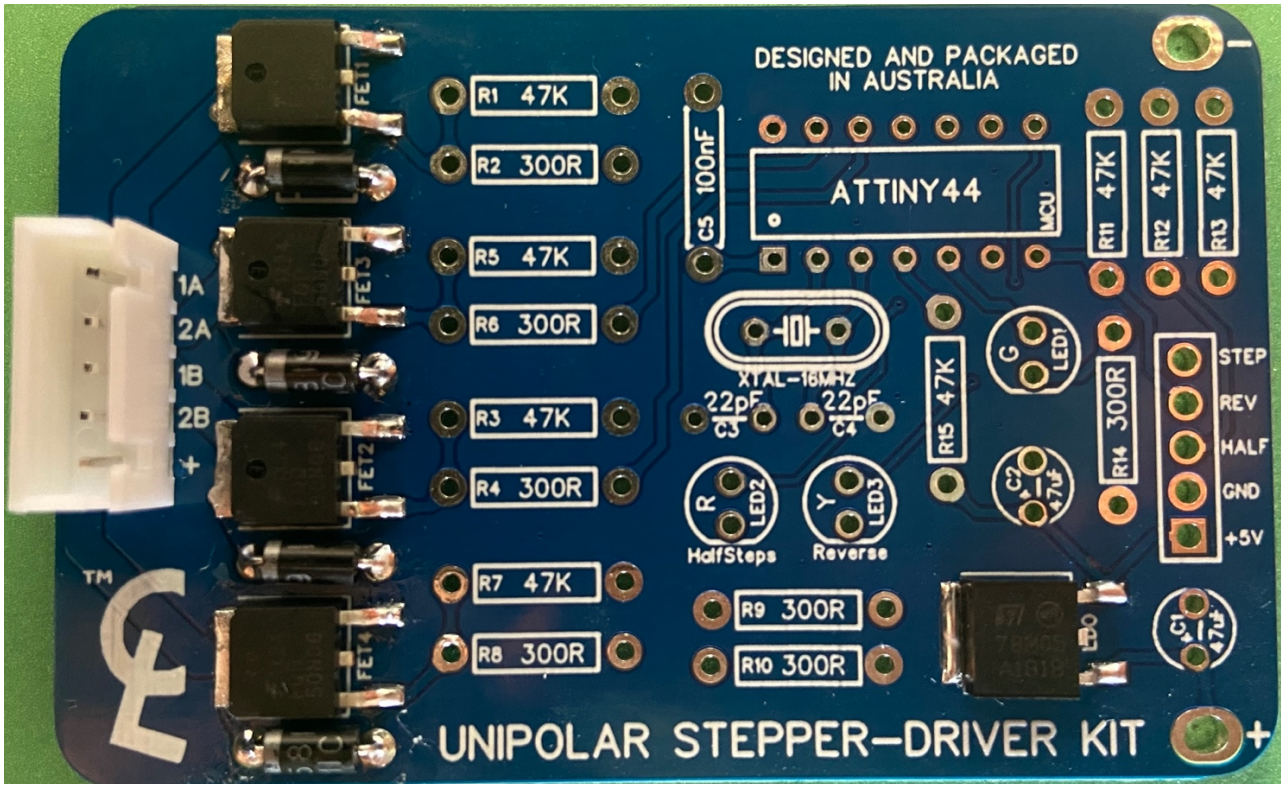
Step-1/Image-1: Solder the FQD50N06 N-Channel MOSFETS to the PCB at the spaces labelled FET1 - 4. It is recommended to lightly tin the large square drain pad before placing the component, once positioned correctly use a flat head screwdriver or a thin piece of tape to hold the component in place while the gate, source and drain connections are soldered. Repeat this for all four MOSFETS, then in the same way connect the L78M05 voltage regulator to the component space labelled LDO.



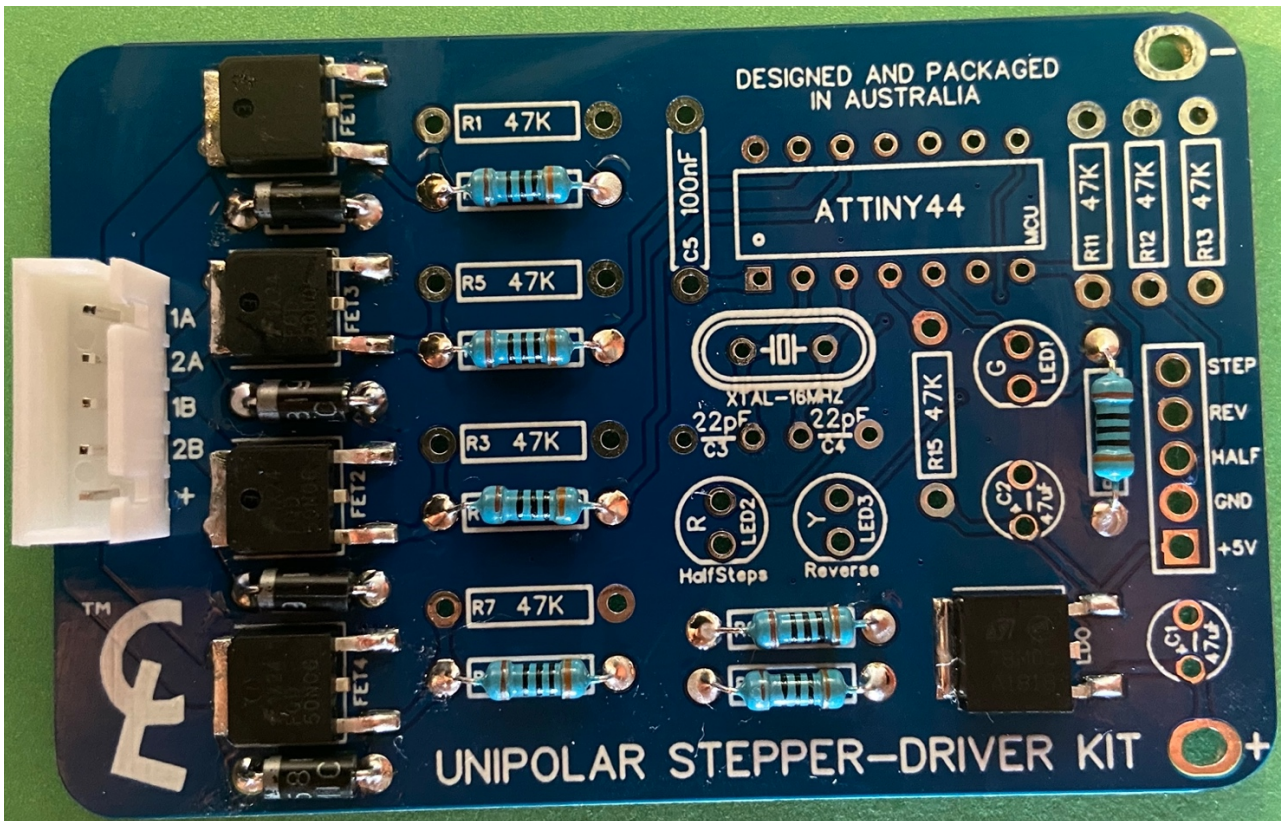
Step-2/Image-2: Solder the 1N5819 Schottky diodes to the PCB at the spaces labelled D1-4. Make sure the diodes are placed with their grey stripe facing towards the motor connector as depicted by the component outline on the boards silkscreen. *This is important as diodes are polarised devices and will cause damage if placed in reverse.*



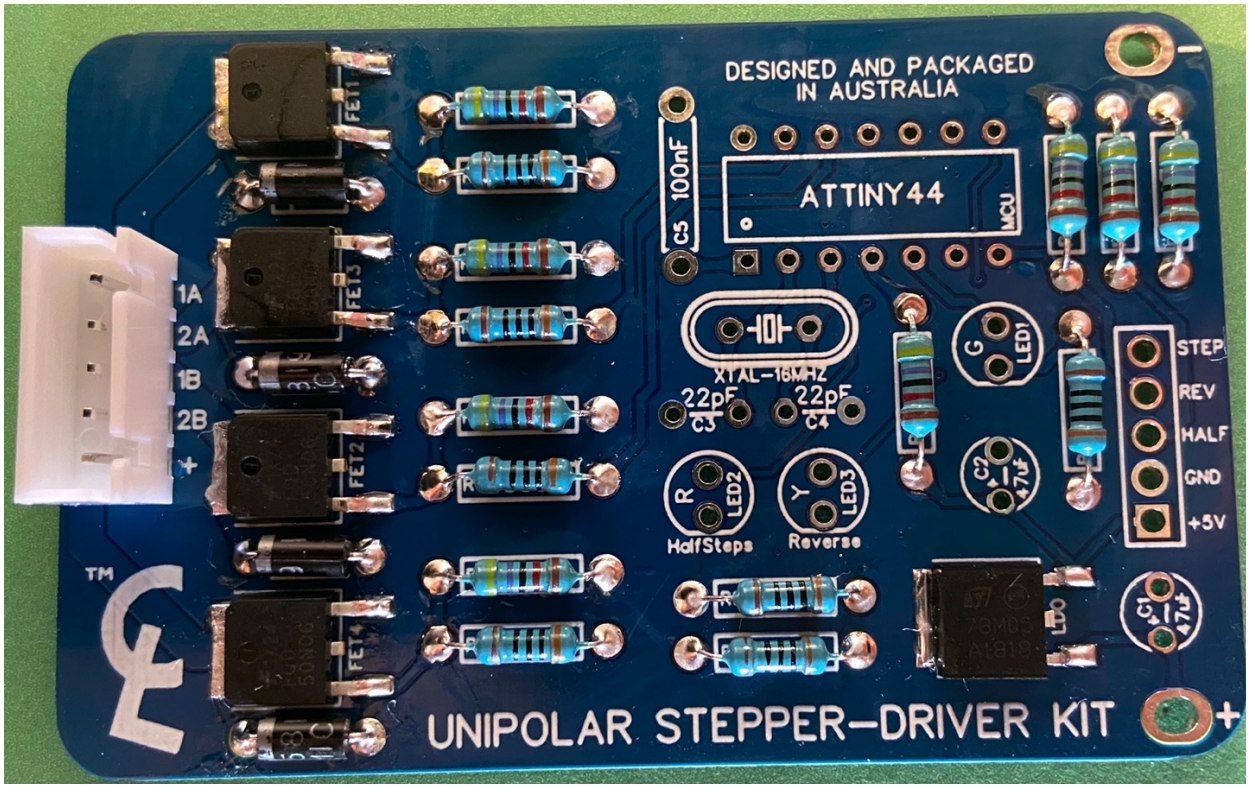
Step-3/Image-3: Solder the 5 pin JST connector to the PCB. Make sure the slots in the connector align properly with the boards silkscreen as the connector only works properly in one direction. It also helps to have a piece of tape hold the connector in place as the solder connections must be made on the underside of the board.



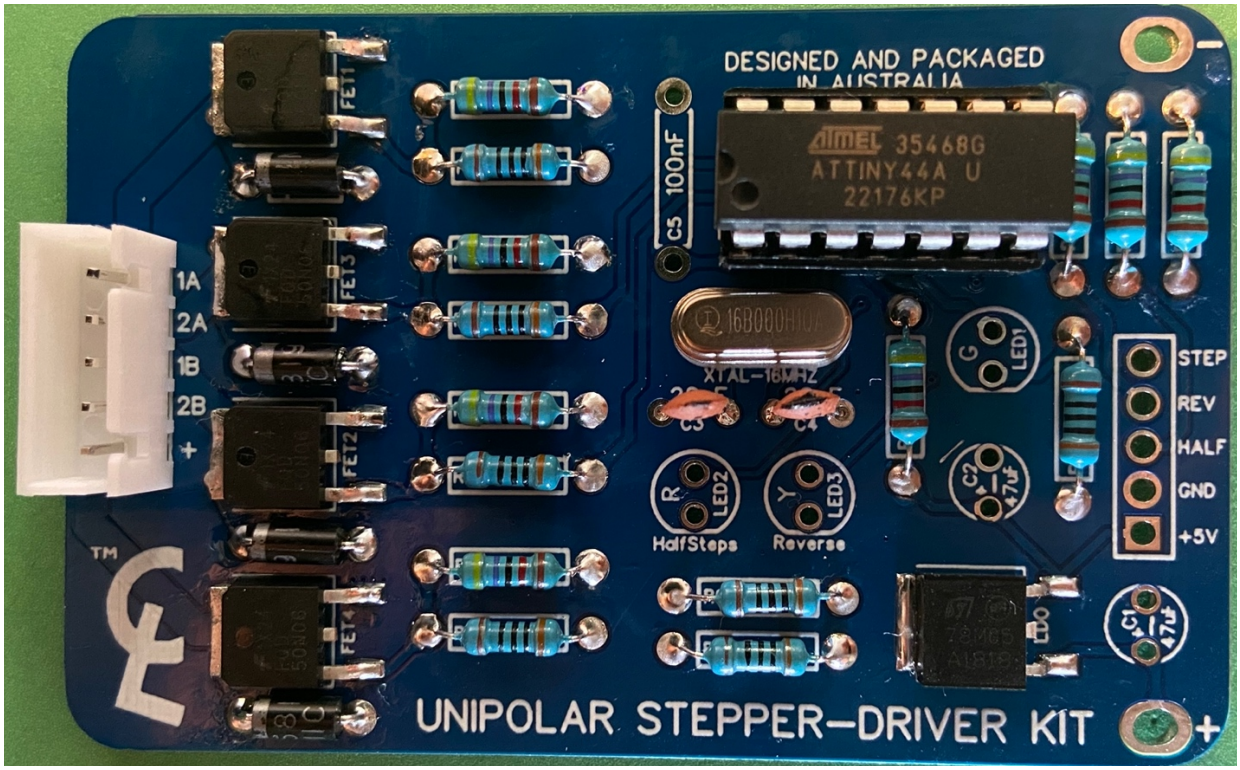
Step-4/Image-4: Solder the 300R *(R = Ohm)* resistors to the PCB at labels R2, R4, R6, R8, R9, R10 and R14. Place the resistors as shown below, solder one end and then check that the resistor is still well positioned, if it is, solder the other end and then repeat. After all the solder connections are made, carefully cut off the extra resistor legs from the board's underside.



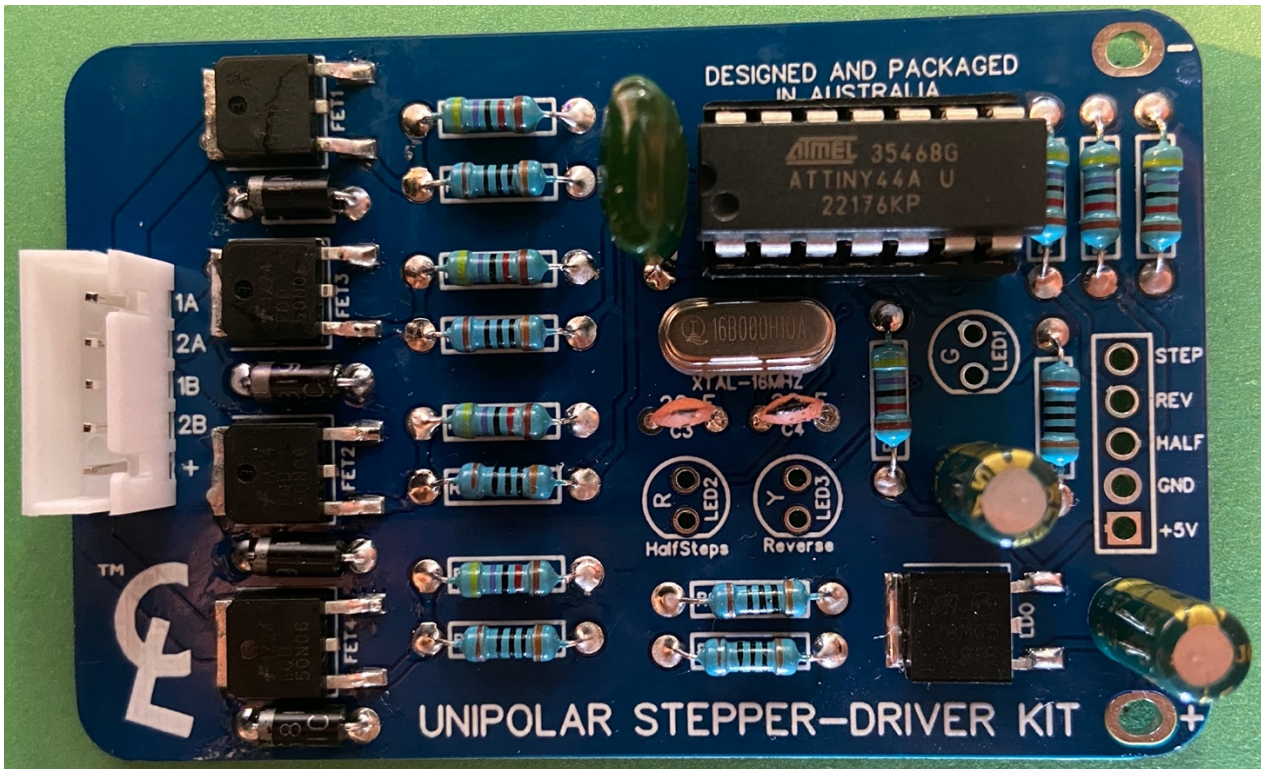
Step-5/Image-5: Solder the 47K Ohm resistors to the PCB at labels R1, R3, R5, R7, R11, R12, R13 and R15. Place the resistors as shown below, solder one end and then check that the resistor is still well positioned, if it is, solder the other end and then repeat. After all the solder connections are made, carefully cut off the extra resistor legs from the board's underside.



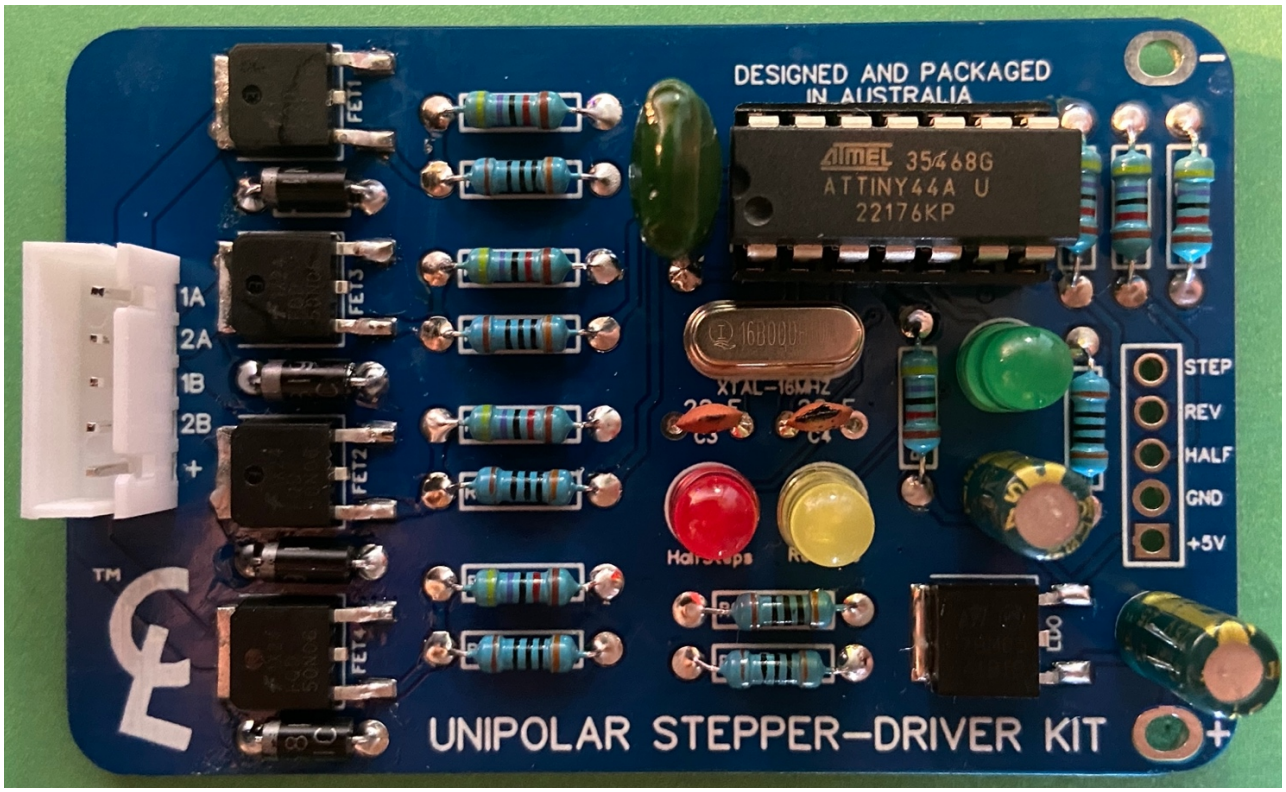
Step-6/Image-6: Solder the preprogramed Attiny44 microcontroller *or a DIP14 socket* to the PCB at the position labelled MCU, then solder the 16Mhz crystal oscillator at label XTAL-16MHZ followed by the two 22pF ceramic capacitors at labels C3 and C4. All connections must be made on the underside of the board, so tape is recommended to hold the components in place while soldering.



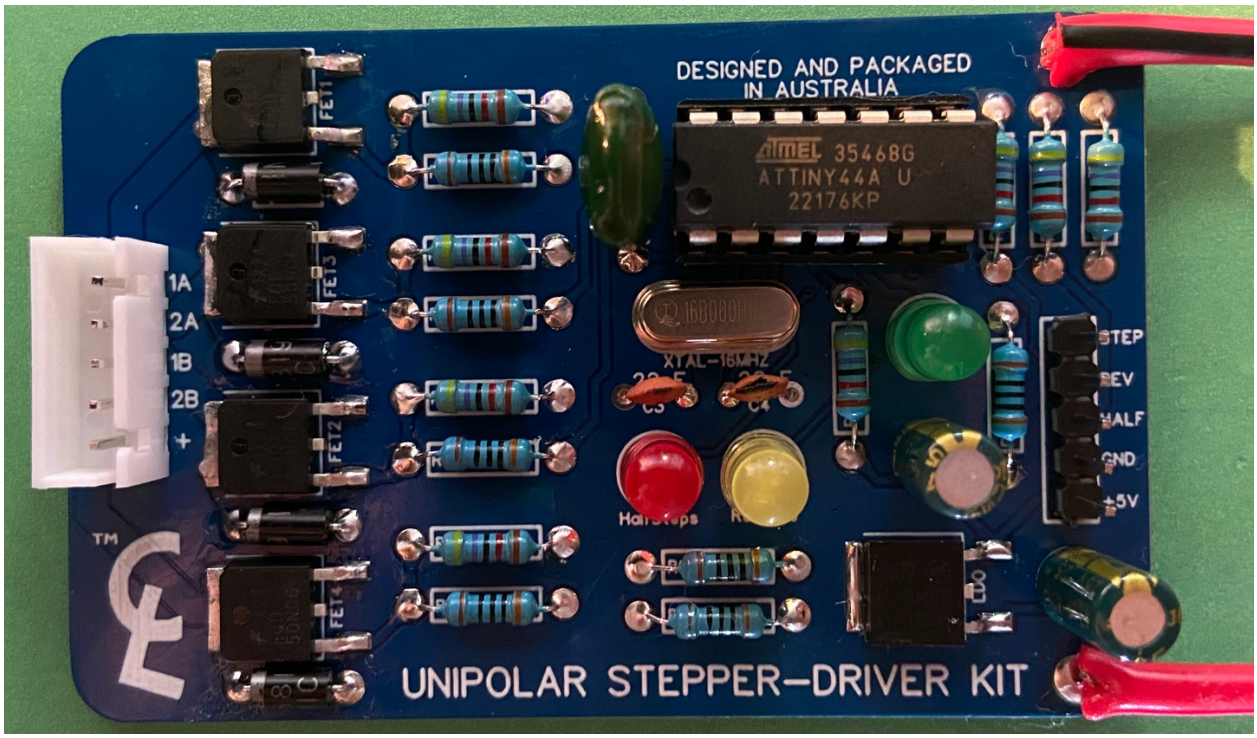
Step-7/Image-7: Solder the remaining power supply and decoupling capacitors to the board. First place the large green 100nF film capacitor into the spot labelled C5. Then place the two 47uF electrolytic capacitors into C1 and C2, *ensuring the longer lead goes through the hole marked with a + as electrolytic capacitors are polarised*. Again, make all solder connections on the PCBs underside utilising tape.



Step-8/Image-8: Solder the three indicator LEDs to the LED1, LED2 and LED3 spaces. *Ensure the flat edge of each LED is aligned with the flat edge on the PCBs silkscreen as LEDs are polarised*. The R Y G markings stand for the correct colour of LED, they will also need to be soldered to the underside of the board while secured by tape.



Step-9/Image-9: Solder the 5-pin input header row and power wires to the PCB. These pins will need to be soldered to the underside of the board using tape to secure them while soldering. The input wires connect to the + and - pads on the board, with between 8 and 18 AWG wire being recommended. *The exposed traces on the underside of the board can also be tinned for high current capability*.



Final Step! Clean any excess flux and any other grime that may have been left behind during the soldering process from the PCB, using a cotton swab or a cloth dipped in isopropyl alcohol is recommended. Ensure all solder connections look good via visual inspection, using a powerful torch helps noticeably with this and then that's it, you have just built your own unipolar stepper motor driver, congratulations!

Driving Unipolar Stepper Motors

Overview: These types of stepper motors are called unipolar as while they are in motion current only flows in one direction, from the centre-tap in each coil *also referred to as phases*, to ground. When the centre tap is powered only the four connections between each phase and ground need to be closed and opened to operate the motor, in this driver's case the switching is done by four N-channel MOSFETS. Driving these types of stepper motors in comparison to their bipolar counterparts will require fewer components and a simpler circuit, making it favourable in low cost and power applications where high levels of torque and speed are not essential.

Drive Sequence: The following tables show the drive sequence used in this kits source code. *1 means the MOSFET/phase should be in an on state (Closed/conducting current) and a 0 means it should be in an off state (Open/not conducting current).*

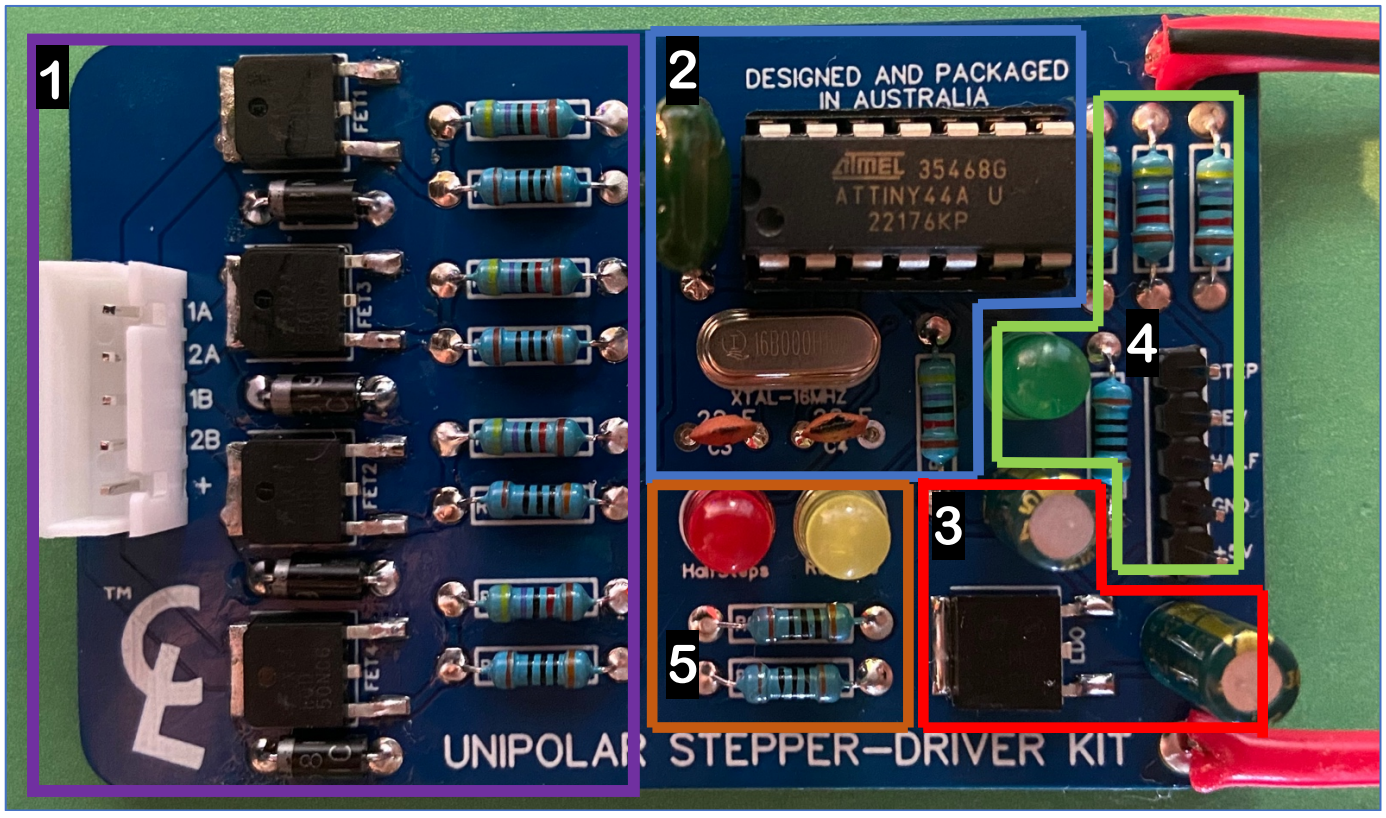
Unipolar Half-Step:

STEPS	FET1/PHASE 1A	FET2/PHASE 1B	FET3/PHASE 2A	FET4/PHASE2B
1	1	0	0	1
2	1	0	0	0
3	1	0	1	0
4	0	0	1	0
5	0	1	1	0
6	0	1	0	0
7	0	1	0	1
8	0	0	0	1

Unipolar Full-Step:

STEPS	FET1/PHASE 1A	FET2/PHASE 1B	FET3/PHASE 2A	FET4/PHASE2B
1	1	0	0	0
2	0	0	1	0
3	0	1	0	0
4	0	0	0	1

Circuit Board Breakdown:



Circuit Explanation.

1) PURPLE: Power stage. The N-Channel MOSFETs with the help of a few other passive components are controlled by the MCU to set each phase as per the drive sequence. A 300 Ohm resistor is placed between each FETs gate and the corresponding IO port, this keeps the current draw at a safe limit, also reducing any unwanted noise on the gate. The 47K Ohm pull-down resistors connect between the gates and ground to help quickly discharge them as the IO port goes low. There are also protection diodes connected across the drain and source path of each FET. This is done to protect against EMF and voltage spikes, as motors are inductive devices, they generate EMF which unless dissipated through the diode, could destroy the FETs.

2) BLUE: Microcontroller. The brain of the device, its core is an Atmel Attiny44 Microcontroller or MCU for short, running on an external clock consisting of a 16Mhz crystal oscillator and two 22pF load capacitors. It also uses a pull a up resistor holding the reset/brown out detect port on the MCU at 5V and a decoupling capacitor to protect the MCU against any noise on the power lines. This gives the MCU everything it requires to function correctly.

3) RED: 5V power supply. The L78M05 regulator and two 47uF noise reduction capacitors create a stable 5 Volt power supply. This supply powers the Microcontroller and the +5V header pin, it is a very simple supply although being a linear regulator, its inefficient. Higher supply voltages into the regulator, or higher current draw on the 5V output, will cause the L78M05 to dissipate heat. This is not a problem when powering the MCU as it draws a low amount of current but be careful connecting more current hungry devices to the +5V pin.

4) GREEN: Input pins. The header pins allow external inputs to be connected, giving users control of the motor's steps, spin direction and the step size. Three of the pins connect to input IO ports on the MCU with the other two at ground and +5V. Each input has a 47K Ohm pull down resistor on it to ensure the pins are never left in a floating state. There is also an indicator LED with current limiting resistor connected directly to the STEP input.

5) BROWN: Indicator LEDs. This visual representation of the drive options that are set via LEDs, is not necessary to achieve motor drive, but has been included for user friendliness. It consists of two red and yellow indicator LEDs connected to the MCU, grounding via 300R current limiting resistors. The LEDs are set to be either solid or blinking, indicating the motors rotation direction and step size, *with forward, full stepping set as default*.

Source Code: The drivers **source code has been preprogramed into the Attiny44** Microcontroller. But for those who are looking to customise this driver for their own application the annotated and Arduino IDE friendly program is made available via the product page at www.casuallyloaded.com as a .ino file. *The code does assume the user has a basic understanding of C++, bitwise operators, direct AVR port manipulation and Arduino variables namely the millis() function.*

END OF DOCUMENT