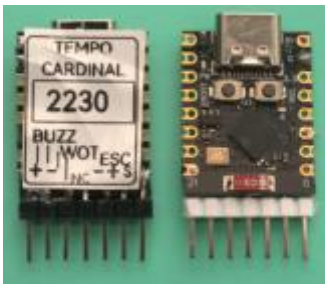
www.dva-controls.com

Tempo Cardinal free flight brushless motor profiler

Since the dawn of aeromodelling, free flight model aircraft have been powered by rubber motors, small internal combustion engines, even rocket motors - but the convenience offered by electric motors is becoming ever more popular. This power method requires some form of timer to control the motor run, and the Tempo device allows the user to define a set of flight parameters for a free flight model aircraft powered by a brushless electric motor. This version is unsuitable for brushed DC motors.



The device is mounted within the aircraft and behaves in a similar manner to a single channel radio control receiver controlling the motor speed - but it should be understood that after the flight has commenced, the user does not have any control over the model (other than an emergency stop).

Instead of adjusting tiny controls on the device, the accompanying Android app is used to define 7 flight phases (including a delay), entering values for motor power, period, and the time to ramp from the preceding power level. To commence a flight, the app sends these values to the Tempo via a wireless Bluetooth connection and the device begins to run the flight profile. Typically, this might proceed thus:

User presses START for 1 second, data sent to Tempo which initiates the profile:
Delay: 10 seconds
Idle: Motor runs at a slow speed for 4 seconds
Take Off: Motor accelerates from the Idle speed to 100% over 4 seconds, runs for 5 seconds
Climb: Motor decelerates over 4 seconds to 80% power, runs for 3 seconds
Cruise: Motor decelerates over 3 seconds to 75% power, runs for 10 seconds
Descent: Motor decelerates over 2 seconds to 40% power, runs for 5 seconds
Landing: Motor slows over 2 seconds to 10%, runs for 10 seconds
Stop

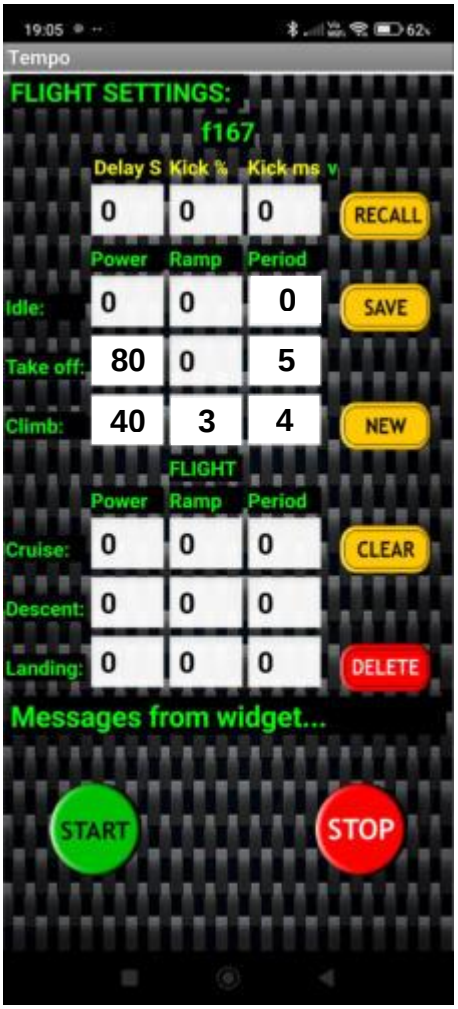
The Tempo must be connected to an electronic speed controller suitable for a brushless electric motor. When used with a 2S battery, the ESC must provide a 5v output (using a BEC). Brushed DC motors may be liable to start running at power up, and should not be used with this version of the Tempo. The device provides the pulses to control the ESC that would otherwise by provided by a radio control receiver.

- Batteries - 2/3S LiPo
- Password protected: Each Tempo has a unique PIN, which must be entered into the app to allow connection.
- Model memories: The app can store the flight profiles for many models.
- Optional beeper: A connection is provided for a 3v lightweight buzzer which will activate some time after a flight, hopefully providing assistance in locating a model flown outdoors.
- Emergency STOP: After commencing a flight, the model will still be connected to the app, albeit over a short range. Should, for example, an indoor model head for the roof, pressing the stop button will terminate the flight.

Trimming assistance - the Tempo may be installed within a model with no need for access - less hatches to worry about. It is also very useful when trimming a new model - all of the flight phases are optional and, for example, you may wish to just set a simple brief, low power run after completing test gliding. This is similar to gradually increasing the turns on a rubber powered model, and you can gradually increase the power and duration of your test flights.

Download the Tempo Profiler app from www.DVA-Controls.com - Android only, sorry....

The Tempo in use:



Please remove the propeller from the motor, until you are familiar with operating the Tempo.

Having installed the device - preferably on a test rig for familiarisation - open the app on an Android phone or tablet.
The app is supplied with a default model - 'COBRA' - and this name will appear at the top of the screen. Although you may experiment with the settings, this model will not be able to connect to the Tempo and cannot be deleted.

Below, are the fields to enter the flight parameters and we will illustrate how to use these with a simple example.

Enter the PIN:

Scroll to the bottom of the screen and enter the 4 digit PIN supplied with the Tempo - this will be appended to the details of the new model that you are about to create.

Click 'NEW' and enter a name for a new model, click 'OK'.

Power levels are entered as percentages, 0-100.
Periods and Ramps are entered as seconds (other than the Idle kick start, in mS).
In the field adjacent to 'Take Off', in the 'Power' column, enter 80. Enter 5 in the 'Period' column.
In the field adjacent to 'Climb', enter 40, followed by 3 in the 'Ramp' column, and 4 in the 'Period' column.

Click 'SAVE'.

Now we need to connect to the Tempo.
Scroll down to the bottom of the screen to find the Bluetooth buttons. Power up the ESC and the Tempo will start up - the blue LED will blink once a second - click 'SCAN'.
After a moment, click 'CONNECT' and a list of Bluetooth sources will be displayed - click on the entry for the Tempo.
Every Tempo will advertise itself with a unique number (unrelated to the PIN), and you should make a note of yours.
Eg: *Tempo1.00_0099*
If all is well, the app will connect to the Tempo - watch the prompts field...and the blue LED will blink twice a second.

When the message 'STOPPED' appears, you're now ready to run a profile. When safe to do so, (check the motor/prop for obstruction) press and hold the 'START' button until it buzzes.
The motor should start and run at a fairly high speed for 5 seconds, before decreasing to a slower speed over 3 seconds. It will run at this speed for 4 seconds before stopping.

Whilst the app is still connected to the Tempo (within range), it will highlight the current flight phase.
Clicking 'STOP' will stop the motor and terminate the flight.

The various flight phases may be defined in a similar manner - other than 'Idle', which is a special case.

Idle phase:
The aim here is to run the motor at a very slow speed, perhaps to simulate a pilot going through their final checks before take-off. However, getting an electric motor to start at low power can be difficult, due to the inertia of the motor and prop. The Tempo tries to work around this by giving the motor a brief kick of high power before settling down to the idle speed
The very top row of parameters are used to define the initial delay, the amount of power for the 'kick start', and the

period. This entry is in milliseconds - entering a value here of 100 would kick the motor for 100/1000 second before dropping to the value defined for ‘Idle’. Some experimentation will be needed for each motor/prop combination - but you can set this up without going for a complete flight, just experiment with the Delay and Idle settings.

NB: To achieve a consistent performance, it is important that the battery be fully charged.

Model memories:

The app can store the details for any number of models, or you may wish to store several flight profiles for the same model. The default model has the name ‘COBRA’ (nods to ancient ELITE players) and you may experiment with the settings. To store any changes for next time, press the ‘SAVE’ button.

When opened, the app should display the model last used.

To create a new model:
Enter the PIN for the Tempo, click ‘NEW’ and enter the new name, click OK to finish. That’s it.

To recall a model - click the ‘RECALL’ button and select.

To delete a model - click the ‘DELETE’ button and select. You can’t delete the Cobra model.

Should you wish to clear all fields to 0, click ‘CLEAR’.

ESC calibration:

When an electronic speed controller is first used with a radio control system, it is usual to calibrate the ESC against the transmitter in order to access the complete speed range. The following is the usual practice - but do consult the instructions for your particular ESC for any differences...

With an r/c setup, the transmitter is switched on, and the throttle stick set to full power.
The receiver and ESC are powered up - the ESC will detect the full power signal and enter its programming mode, usually signified by one or more beeps. The throttle stick is moved to the closed position, the ESC beeps - and may then be disconnected, having set the travel limits for this transmitter.

The Tempo can simulate this method. REMOVE THE PROP!

Connect the Tempo to the ESC /motor, don’t apply power yet.

Using the provided jumper, link the 2 pins shown here.

Power up the ESC - the Tempo will output a ‘full throttle’ signal which will force the ESC into programming mode.

When you hear the beeps, remove the jumper link and power down. Your ESC should now be calibrated against the signal range output by the Tempo.

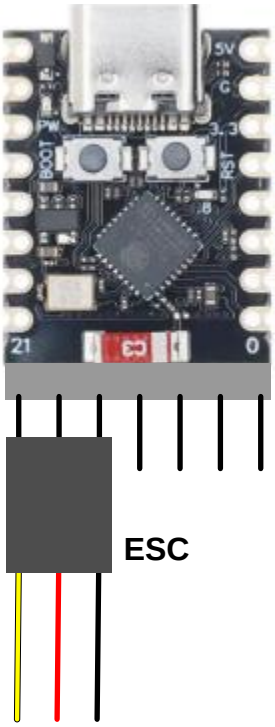
Optional beeper:

A 3v active beeper may be connected to the pins shown here, noting polarity - the beeper will have a + mark and you should ensure that it is connected correctly. To activate the beeper, enter a number into the field in the app - the beeper will activate this many minutes after a flight, and will emit 2 short beeps every 20 seconds.

Battery considerations:

The Tempo doesn’t monitor the battery voltage. It is highly recommended that the battery be fully charged before each substantial flight in order to achieve consistent flight patterns. Please be sure to use batteries of known good quality and ‘C’ rating.

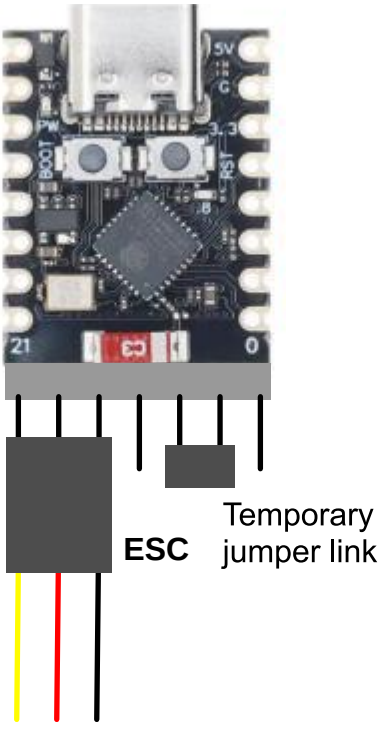
Electrical connections:

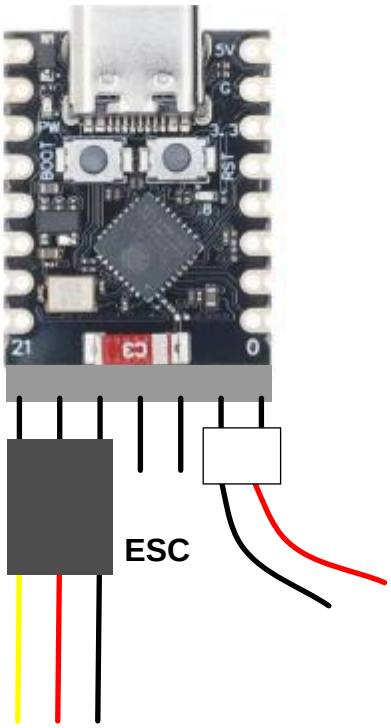


For 2/3S LiPo:
The ESC must have a battery elimination circuit (BEC) providing a 5v output. Simply connect the cable from the ESC to the Tempo as shown. The signal wire - this may be white, yellow, sometimes orange - goes to the outer pin (21).

ESC calibration: REMOVE THE PROP!

To calibrate your ESC against the Tempo, connect these 2 pins with a jumper link before powering up. The ESC will detect a full throttle signal and enter programming mode, usually signified by one or more beeps. Remove the link, wait for the beep and power down. See your ESC guide for details.





Optional buzzer:

Connect the buzzer as shown - please be careful to get it the correct way, or the buzzer may be damaged. You may find that the sound is louder if the buzzer is fixed to a sheet component - don't forget to remove the sticker!.

Optional beeper

If you've ordered the optional beeper kit, you will find:

Beeper module
Cable with connector
Heat shrink tubing

You will need to solder the cable to the beeper module, noting that both are polarised - solder the red wire to the + connection on the module, the black wire to the - module wire. The cable may be shortened if desired. Remove the sticker from the module.

Don't forget to slip a short length of heat shrink into place before soldering.....

In use, push the connector onto the Tempo pins, noting polarity as shown above.

Specifications:

Device name: Tempo Cardinal*
Microcontroller: ESP32c3 Supermini
Supply Voltage: 5v - provided by the ESC which may be powered by 2S or 3S LiPo.
Weight: 2.7g
Power consumption: 30mA when connected to BLE
Radio link: Bluetooth Low Energy (BLE)

* Why 'Cardinal'?

In our winter indoor flying sessions, I'd seen people using electronic profilers with electric FF models, but thought that a remote app based version might work well. The Tempo is based on an ESP32C3 microcontroller board, one of the smallest available with built in Bluetooth, and the code written in the Arduino IDE over a long development period.

The Tempo app was written using the excellent online Android editor, MIT App Inventor - highly recommended for anyone wishing to have a go at Android apps. Although I do have a programming background, I'm not a 'proper' Android developer - but have been using the the MIT editor for several years, and it is possible to develop some surprisingly complex apps here.

Various obstacles were overcome whilst developing the widget, and eventually the Tempo was ready for its first flights outdoors. The model chosen was the classic Veron Cardinal - hence the name for this variant of the Tempo. This has a problem with driving DC brushed motors, and a future version will feature a daughter board with an ESC suitable for small DC brushed motors.

Constructing the Tempo Cardinal

Building a Tempo from this kit is not difficult for anyone who can use a soldering iron with confidence. You will need:

- Good quality, fine tipped soldering iron, preferably one that can reach high temperatures
- Solder appropriate for copper, preferably fine gauge with a flux core
- Small clamp - crocodile clip or similar - I find that a wooden clothes peg is useful for holding small items
- 5 minute epoxy, UV resin, or similar
- Nail varnish - gotta look cute for this job.....

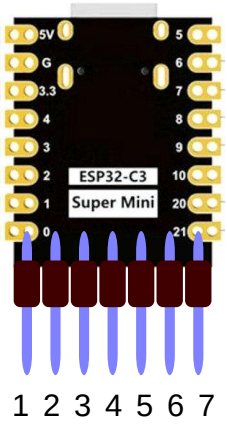
The connecting wires are enamel coated copper. The enamel is designed to burn off when soldered, but it does need to be hotter than usual when soldering. I've found that a temperature of 350, with a little solder on the tip, will do the job. Wipe off any debris before soldering the wire in place. If you find that your iron is not hot enough to do the job, the enamel can be scraped away with a sharp knife before tinning.

Cut 2 lengths of 25mm, one length of 15mm, and one of 30mm, from the copper wire. For each one, tin 4mm of both ends with a hot iron.

Take the strip of header pins (trim to 7 pins if longer) and clamp so that the row of shorter pins are handy - apply a little solder to each, but don't linger with the iron (a lower temp may be used here) - if the pin becomes too hot, it may melt the plastic. Don't tin the longer pins.

Now we will solder the wires to the header pins in the following order:

PIN	LENGTH
2	25mm
3	15mm
5	25mm
6	30mm



The wires should all lie on the top of the pins if possible - take care with this job, it can be tricky holding the wire in place and applying a momentary touch with the iron.
Give the wire a gentle tug, to confirm security.
Avoid getting a blob of solder on the pin, as they will need to lie flat against the PCB. A spare set of header pins and wire is included, should you need them.

Take the PCB and, noting the orientation here, clamp the row of header pins to the underside, at the opposite end to the USB connector. The short pins should be overlapping the PCB and the plastic strip abutting the edge of the PCB. A wiring diagram will be found below.

Check that the endmost pins are touching the contacts on the PCB - they are marked '0' on the left, and '21' on the right. When everything is correct, solder these 2 pins to the PCB contacts. They will hold the header strip in place while we add the wires.

Take the wire on pin 2 and thread the other end through the hole marked 'G' - this is the ground connection. Ensure that the wire is flat against the PCB, but don't solder it yet.

Take the wire on pin 5 and thread the other end through the same hole marked 'G'.
Check that the tinned ends of both wires are in the hole and apply a little solder to secure.

Take the wire on pin 6 - this is the 5v connection from your ESC. Thread the other end through the hole marked '5V' and solder as before, noting the caution found in the wiring diagram.

Take the wire on pin 3, thread the other end into PCB hole 2 and solder in place.

The electrical work is now complete. The next step is to secure everything - but you might want to test your creation beforehand to check your soldering. Handle the Tempo carefully, as the unprotected wires are vulnerable, and try using it as described in the user guide - preferably on a simple test rig rather than installed in a model.

Finishing off:

The header pins and wires need to be secured and protected.

Mix a little 5 minute epoxy or apply a little UV resin (this is my preference) across the header pins overlapping the PCB, building a long bead across the pins. Allow the epoxy to cure, or zap the resin with a UV source.

When set, apply nail varnish across the copper wires, brushing across, rather than along, so as to build a fillet around the wires, immobilising them.

That's it! I hope that it wasn't too difficult, and that your Tempo gives you many happy flying hours.