# Casio PRO-101 TC5006P RAM Replacement

This modification to the Casio PRO-101 replaces the TC5006P RAM ICs with an RP2040 microcontroller. The TC5006P devices are used to store the program steps of the calculator. They are low power devices and can be powered by three button cells when the main circuitry of the calculator is powered off. This provides the continuous memory function of the PRO-101.

The RP2040 is a dual core ARM based microcontroller. It is not a low power device so storage of program steps cannot be done using the three button cells as the original circuitry did.

Program Memory

The PRO-101 has 256 steps. Each step is stored in a byte. The TC5006P is a 1024 x 1 bit device. It can store 128 bytes and so two devices are used to store the 256 steps. There is little information about the T5006P to be found, but it turns out that the later TC5508P device has the same pinout. A datasheet is available and the pinout is shown here:
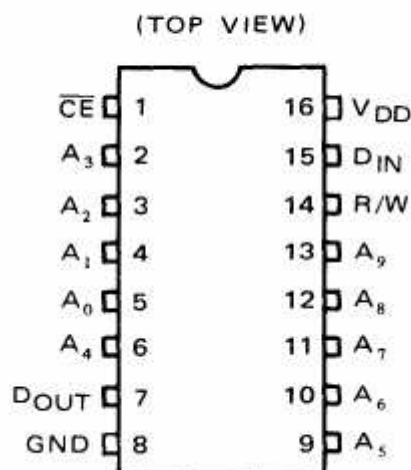
The device has a data in signal and a data out signal, whih is not unusual for devices made in the 70s and 80s. There are address lines, a R/W signal and a chip select. The RP2040 reads the address lines, CE line, data in line and R/W line to work out what data to put on the data out signal, and when. It monitors both of the CE signals that went to the two original chips and emulates both of them. This isn't difficult as they are on the same bus and are never accessed at the same time.

The RP2040 has 264K of RAM and so can easily emulate the 256 bytes of program space. The RP2040 RAM is accessible by both ARM cores, but simultaneous access can cause contention and stalling of a core.

Hardware Modifications

The hardware of the calculator has been altered as little as possible. The two TC5006P chips have been removed as they were faulty and could no longer store programs. One of the TC5006P footprints has been fitted with a custom PCB with an RP2040 Zero type module. There is a flying lead from the second TC5006P footprint to provide the second CE signal.

The RP2040 Zero is powered either from the PRO-101 TC5006P supply or from USB. The modified unit does not use the button cells and they should not be fitted to this modified calculator.

The power consumption of the modified calculator is higher than the original by a factor of about two to three. I have used rechargeable AA batteries in the calculator for many hours while developing the code for the RP2040, so this calculator is still portable.

Interaction with the code running on the RP2040 is either through the calculator keyboard (details in another section) or a USB command line interface (CLI). There are not very many ways to get the USB cable out of the calculator. I ended up removing the A adapter socket and running the cable through the resulting hole. The battery compartment couldn't be used to run the cable out as the battery cover couldn't then be fitted and the batteries tended to fall out. A wireless (Wifi) interface was prototyped but the module used couldn't be powered by the calculator. The power in the calculator is quite complicated and no easy, safe way to run the Wifi or Bluetooth interfaces could be found. I didn't want to run the risk of damaging the power circuitry of the calculator. The Wifi interface code is still in the source, but compiled out at the moment.

Memory Emulation

One of the RP2040s performs the emulation of the TC5006P. It has interrupts turned off as it has to respond to the signals on the bus promptly. It does not perform any task other than RAM emulation.

The second core performs all of the interface functions for USB, Wifi and the calculator keyboard interface.

The second core (core0) accesses the emulation RAM of the emulation core (core1) very carefully. If it reads the RAM in bursts then core1 can be stalled and corruption of the emulation RAM can result. For this reason there can be a delay in the user interface before changes to the emulation RAM are seen. This is of the order of 2 or 3 seconds. This must be borne in mind when interacting with the RP2040.

Flash Storage

The same flash memory of the RP2040 that is used to store its firmware is used to store programs for the PRO-101. Flash is non volatile so these programs will still be stored even with power off and no batteries in the calculator.

The RP2040 Zero has 2M bytes of flash, 1000K bytes of this is used to store programs. There are 250 4K slots that can hold a program and a text label.

Emulation memory can be stored in one of these slots and recalled later. Labels for the slots can be set up over USB.

Programs can be saved and loaded to and from flash using keyboard commands, but that does involve using a few steps as a command. To save and load the full 256 bytes the USB interface has to be used.

Flash slots can be erased if needed. There is no way to erase all of the slots in this version of the firmware. Updating the firmware itself does not erase the data stored in the flash slots.

USB Interface

The USB interface starts with a sign-on message and is then menu driven. The sign on is here:

```
/-----------------------------\
| Casio PRO-101               |
| TC5006P Pico                |
/-----------------------------/

Setting GPIOs...
Initialising UART RX...
Starting PIO UART RX interrupt...
UART RX running...
UART TX initialised...
```

This is a list of parts of the firmware as they initialise.

After the sign on message, press either 'h' or '?' for the menu help display:

```
h: Serial command help
?: Serial command help
s: Save emulation RAM to flash slot
u: Send text to uart
I: Information
l: Load emulation RAM from flash slot
z: Zero parameter
E: Erase program slot
D: Display program slot
S: Display slot buffer
d: Dump RAM
L: List Programs
#: Set Slot Number
A: Set Address
a: Display All Programs
T: Toggle Write Protect
W: Write Byte
t: Enter Text Parameter
B: Label Slot
b: List Slot Labels
@: Display trace buffer
!: Boot to mass storage
```

Each of the other options is described below:

s: Save emulation RAM to flash slot

Saves the emulation RAM contents to the flash slot number 'parameter'. The parameter value can be set up by typing numeric characters at the menu prompt. It's current value is displayed in the menu prompt. The parameter can be cleared using the 'z' command.

This command should be issued 2-3 seconds after any changes have been made to the emulation RAM.

u: Send text to uart

This sends the text parameter to the UART. It can be used to send commands to a wireless module (Wifi and/or Bluetooth) if it is fitted. This is not used at the moment.

I: Information
This option displays various information about the firmware.

l: Load emulation RAM from flash slot
This option loads emulation RAM from the slot whose number is in the numeric parameter.

z: Zero parameter

Sets the numeric parameter to zero.

E: Erase program slot
Erases the flash slot whose number is in the numeric parameter.

D: Display program slot

Displays the flash slot whose number is in the numeric parameter.

S: Display slot buffer

This displays an internal buffer.

d: Dump RAM

This displays the emulation RAM as hex and also as keystrokes.

L: List Programs

Displays the emulation RAM as keystrokes.

#: Set Slot Number

A: Set Address

Sets the Address parameter to the value of the numeric parameter.

a: Display All Programs
T: Toggle Write Protect
W: Write Byte
t: Enter Text Parameter
B: Label Slot
b: List Slot Labels
@: Display trace buffer

!: Boot to mass storage

Boots the RP2040 and puts it into mass storage mode, ready to accept a UF2 firmware file. Once copied the new firmware is flashed and the RP2040 reboots.

Calculator Interface

Wireless Interface