

MAX2870 Synthesizer Control

Andy Talbot G4JNT Jan 2023

Overview

The MAX2870 Synthesizer is a fractional-N unit with internal VCO and output doubler that can generate frequencies between 23.5MHz and 6GHz. The chip has appeared as an evaluation board available from several sellers.

The control system is made up of two parts. A small PIC module drives the SPI lines of the chip. The register values are sent to this module on a serial RS232 type interface as hexadecimal values in an ASCII text based command structure. Any terminal programme such as Putty can be used to set the register values which can be stored to the PIC's internal EE storage for immediate boot up.

The controller module has been designed as a postage-stamp sized PCB that plugs directly onto the 10-pin connector on the evaluation boards. The same module (with different PIC firmware) can be used in the same way with the ADF4351 and ADF5355 boards that have a 5x2 way header; the connector pinout is the same. A resistor has to be added to pull-up the CE line (device pin 4)

Operating in conjunction with this module is a Windows programme **MAX2870.EXE** that allows the wanted frequency and other user-settable parameters to be defined. The register values are calculated and can be sent to the PIC module, and optionally programmed into its non-volatile EE memory. The programme can be initialised from the EE contents to save having to completely set up if just small parameter changes are wanted. Major Settings are saved when the programme is closed. Alternatively, instead of direct control the register values can be exported in a format for copy-and-paste into PIC assembler for any other controller where the data is stored in individual byte format.

PIC Module

Figure 1 shows the circuit diagram for this and is identical to a version used for other synthesizer modules from the 'JNT stable. The basic design can be seen at http://g4jnt.com/Serial_SPI.pdf . A PCB allows direct plugging onto the ADF43xx, ADF53xx and MAX2870 Evaluation Modules that have a ten way header organised as two rows of 5 pins. It automatically picks up its 3.3V supply from the board. The PCB fitted to one such synthesizer can be seen in Photo 1. Note the additional resistor needed to pull up the CE line.

Use a terminal programme such as Putty. When the module is first turned on a display such as that in Figure 2 will be seen, showing a brief help menu. The EEprom contents are sent to the synthesizer chip at boot up.

Any register can be changed by entering R followed by the register contents as an eight digit hex number, followed by [rtn]. The last character of the eight is the register address so the complete value is always stored in the correct register. Typing **R2000E729 [rtn]** Sets register 1 to hex 0x2000E729 There is no echo-back of typed characters but the controller responds with an acknowledgement if the command is successful. To store this single register to EE non-volatile memory, enter **W [rtn]** after the response to the register write has appeared. Each register has to be written to EE separately, so six W commands need to be issued to set all registers. To start again, (for example if everything goes wrong), the chip can be reinitialised to the latest EEprom contents with the command **INIT [rtn]** . The response is the same as when first turned on.

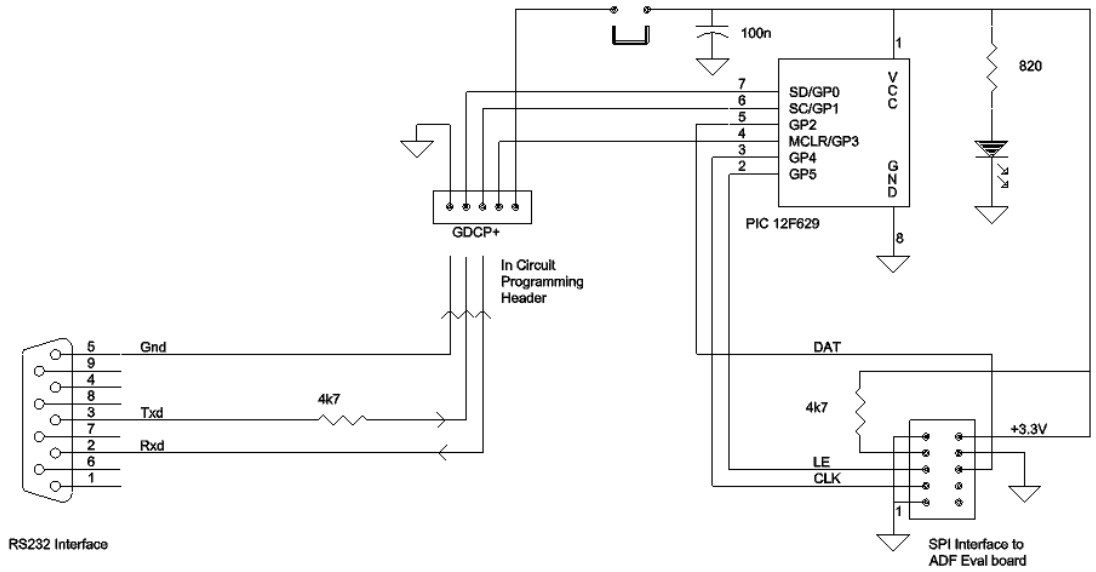


Figure 1 PIC module circuit diagram

```

COM1 - PuTTY
G4JNT
MAX2870 Control
Rxxxxxxxx [CR]
Sxxxxxxxx [CR]
[W]rite reg EE
[I]nit from EE
Uyyyyyyyyy [CR]

0000 0000 0100 0000 0000 0000 0000 0101 0x00400005
0110 0011 1011 1110 1000 0001 0010 0100 0x63BE8124
0000 0000 0000 0000 0000 0000 0000 1011 0x0000000B
1111 1001 0000 0000 0101 0000 0100 0010 0xF9005042
0010 0000 0000 0000 1110 0111 0010 1001 0x2000E729
0000 0000 0010 0010 1001 0100 1011 0000 0x002294B0
Udata {100 MHz }

Init
0000 0000 0100 0000 0000 0000 0000 0101 0x00400005
0110 0011 1011 1110 1000 0001 0010 0100 0x63BE8124
0000 0000 0000 0000 0000 0000 0000 1011 0x0000000B
1111 1001 0000 0000 0101 0000 0100 0010 0xF9005042
0010 0000 0000 0000 1110 0111 0010 1001 0x2000E729

```

Figure 2 Screen Dump from Putty

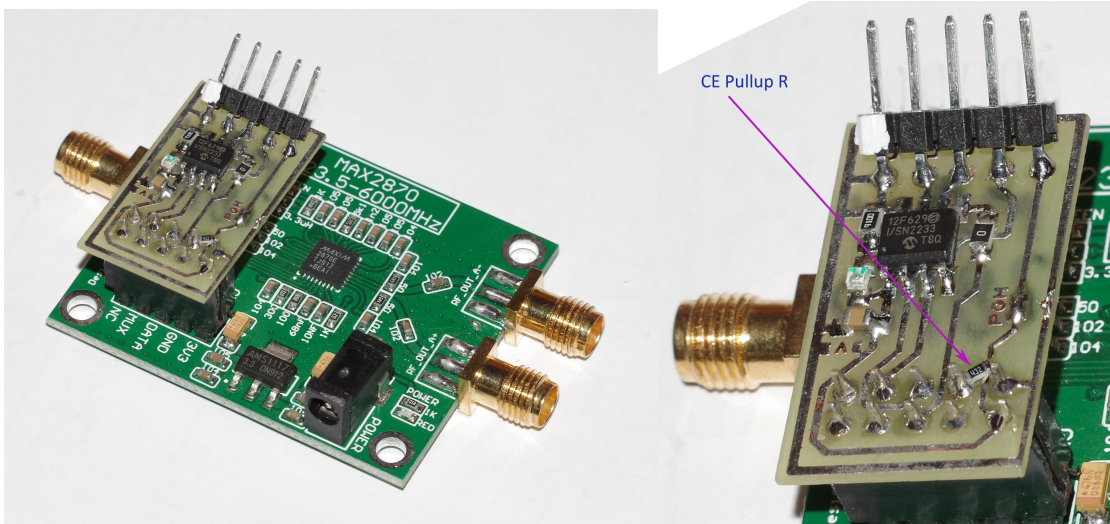
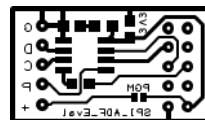
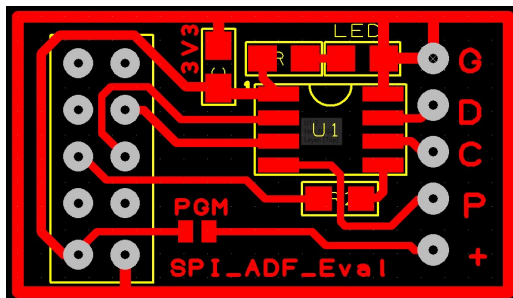


Photo 1 PIC + MAX2870 module PIC module in closeup, showing additional CE pullup resistor for MAX2870



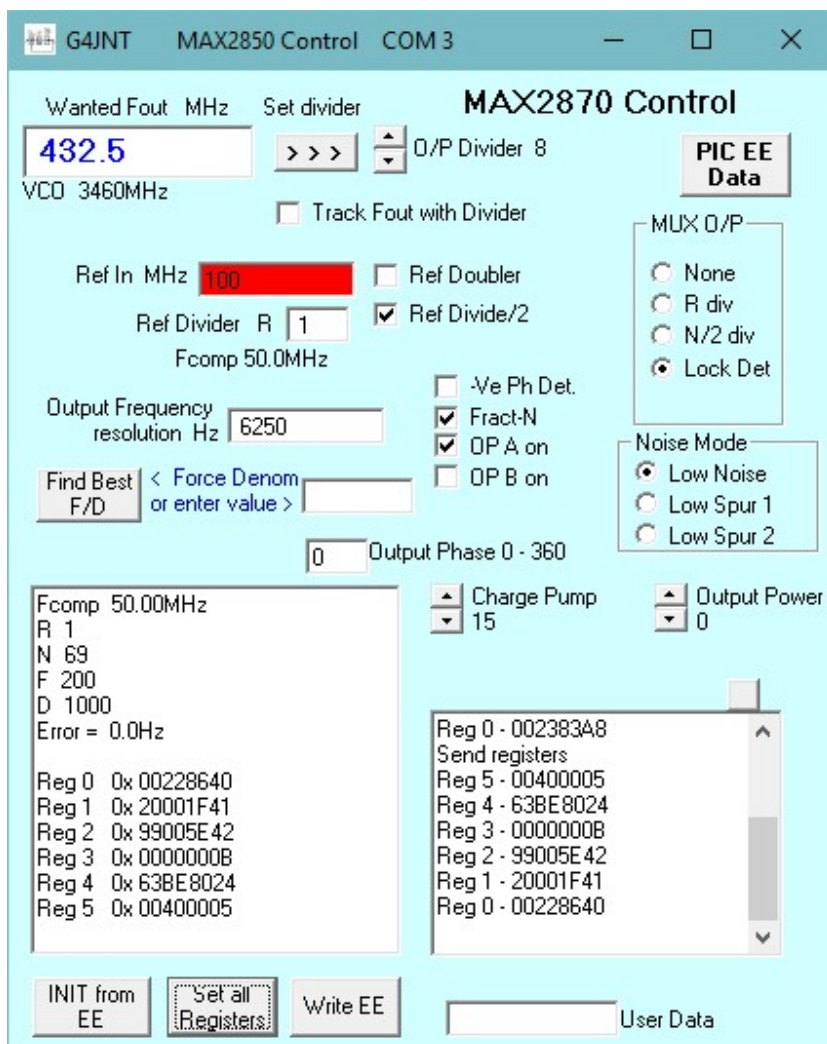
PCB Layout with 1:1 Mirrored artwork

Control Software

For a complete understanding of how to use this synthesizer chip optimally, obtain a copy of the data sheet from <https://www.analog.com/media/en/technical-documentation/data-sheets/max2870.pdf> or the website of one of the suppliers.

To calculate register values for any output frequency, reference input, and for setting most of the important parameters, the software **MAX2870.EXE** can be used in one of several ways. The main user screen can be seen in Figure 3. The user-entry for frequency setting are into the windows at the top. If any of the settings result in a VCO frequency outside the allowed range of 3-6GHz, this is highlighted in red with a warning. If checked, the **Track Fout With Divider** box changes the wanted output frequency when the output divider setting is changed, maintaining a fixed VCO setting.

The reference frequency should be self-explanatory. As this Fractional-N engine can always generate tiny step sizes, far lower than usually needed, it is worth keeping the frequency into the phase-frequency detector, F_{COMP} , as high as possible to reduce phase noise. The frequency doubler *Ref Doubler* can help here if a low frequency input is used such as a 10MHz reference. Unless there is a very good reason to do



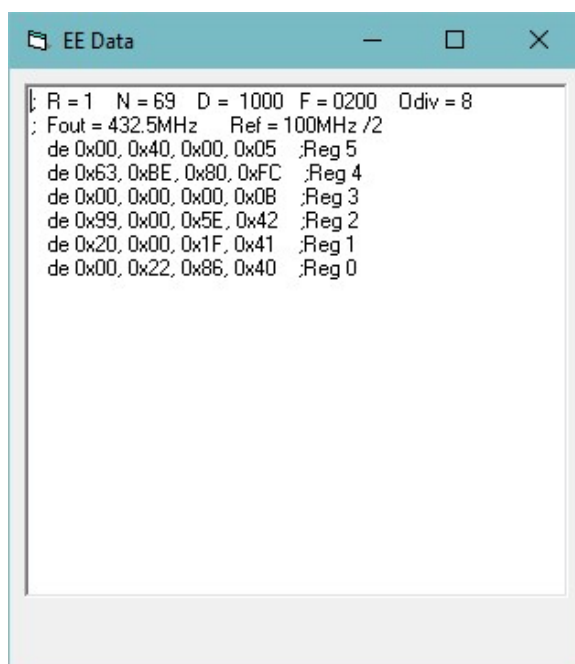
otherwise, set R, the reference divider, to 1. The reference divide by 2, *Ref Divide2* can help phase noise and is useful when the on board 100MHz oscillator supplied with some modules is used. See the data sheet for more details.

The Fract-N checkbox affects only those bits in the control registers specific to Fract-N or Integer operation. Unchecking this to select Integer-N operation does not change any register values – N, F and D will have to be set manually.

For simplicity, *Output Power* adjusts both A and B outputs to one of the four possible levels, Both outputs are enabled and deliver the same divided down frequency.

The calculated register values appear in the info box. The other settings, values, and tick boxes will often need to be changed to optimise various parameters like phase noise and spuri. Consult the data sheet to see how to set these.

Figure 3 Screen Dump from *MAX2870.EXE*



For export to byte storage, click the 'PIC EE Data' button to get an output like that shown. The contents can be copied to the clipboard for pasting into any assembler file.

For communication with the PIC controller module use the selection box at the bottom that initially appears with a red background. It will have been filled with all valid COM ports found. Select the one to be used and click the 'Set COM Port' button. A message box will appear to show the COM port has been opened successfully and the three buttons at the bottom will then be exposed.

'Init from EE' issues an 'INIT' command to the controller that loads the registers from EE. The data fed back on the serial interface is decoded and the resulting frequencies, values and checkboxes updated. The frequency calculated from the read-back data assumes the reference input has

already been set to the correct value.

The values read back can form a good starting point for whatever subsequent changes are needed. When any new values have been entered, click the 'Set All Registers' button to send the complete register set to the synthesizer chip. The update will take a couple of seconds as the bidirectional serial communication cannot be hurried. When you are finally satisfied all is as it should be and the results can be saved, click the 'Write EE' button – this also takes a few seconds to complete. Responses back from the PIC controller appear in the bottom right text box.

Calculating MAX2870 Register Values

The Fractional-N register settings used in the MAX2870 is essentially the same as that for the popular ADF4351. The main differences being that the VCO tunes over a different range of 3 to 6GHz, and the output divider includes a /128 setting, allowing frequencies down to 23.5MHz

The output frequency is determined from :

$$F_{OUT} = F_{COMP} * (N + R/D) / OP_{DIVIDER} \quad \text{where} \quad F_{COMP} = F_{REF} / R * REFDBL / RFEFDIV$$

F_{COMP} should ideally be set to as high a value as possible subject to the limits shown on the data sheet.

The fractional denominator D has a maximum value of 4095 which limits the step size or output tuning resolution to $F_{COMP} / D / OP_{DIVIDER}$. To get any obscure frequency, or as close to it as possible, values for F and D can be selected for minimum frequency error. A value for D can be entered manually or the **MAX2870.EXE** software contains a utility that searches for the best F/D ratio for any specified output frequency after F_{COMP} and $OP_{DIVIDER}$ have been defined. This is activated by clicking the button "**Find Best F/D**". Any residual error is shown in the text window, if it is more than 1Hz the text appears in red.

Software, PIC code as .ASM and .HEX files and higher resolution copies of the diagrams can be found at

<http://www.g4jnt.com/MAX2870.ZIP>

Registering MSCOMM32.OCX

If an error message appears that mentions this file when running **MAX2870.EXE** for the first time, it will be necessary to register this serial port driver. The registration will be needed if you have no already-installed Microsoft Visual software that uses the serial port

Use the copy supplied in the .ZIP, or search the Web and download **MSCOMM32.OCX** which can be found on Microsoft sites and in many other places. Copy the file to any convenient folder / directory.

In a Command screen, go to this folder and type **REGSVR32 MSCOMM32.OCX [rtn]** whereupon it will install for perpetuity.

There are other ways to install this file – a search engine will reveal alternative solutions.