ADF5355 Synthesizer Control

Andy Talbot G4JNT Sept 2018 UPDATE March 2020

Overview

The ADF5355 Synthesizer is a fractional-N unit with internal VCO and output doubler that can generate frequencies between 54MHz to 13.6GHz. The double-modulus Frac-N architecture means that even at the highest output frequencies a resolution of a small fraction of 1Hz can be achieved. The chip has recently become popular as an evaluation board available from several sellers. Note that additional decoupling of the regulated supply can help spurii and phase noise. Full details of the necessary modifications, written by Bryan Flynn GM8BJF, are in the December 2017 edition of *Scatterpoint* .

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 ADF 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 used in the same way with the ADF4351 board that has a 5x2 way header; the connector pinout is the same.

Operating in conjunction with this module is a Windows programme 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 new PCB has been developed to allow direct plugging onto the ADF43xx and ADF53xx 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. For other versions of ADF5355 evaluation modules with different header pin layout, the original PCB can be used with the pins cross-linked.

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 and the EEprom contents 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 **R56060002 [rtn]** Sets register 2 to hex 0x56060002. 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. The bottom of Figure 2 shows the successful execution of a couple of register updates and an EE write.

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

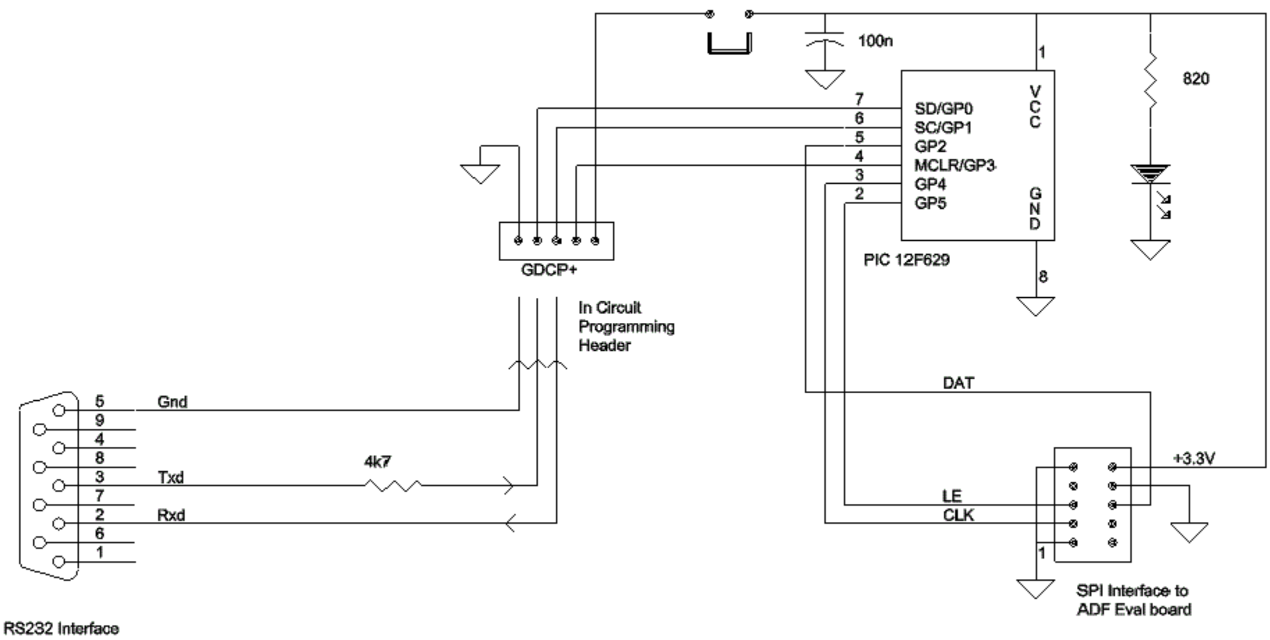
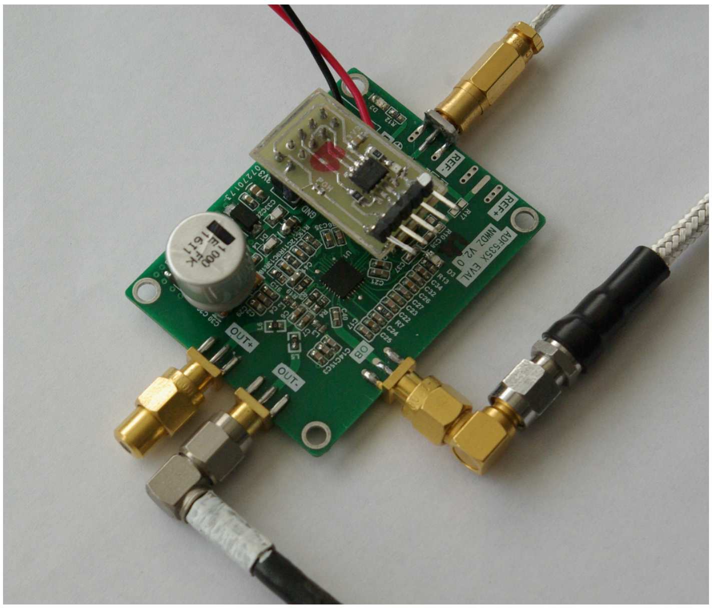
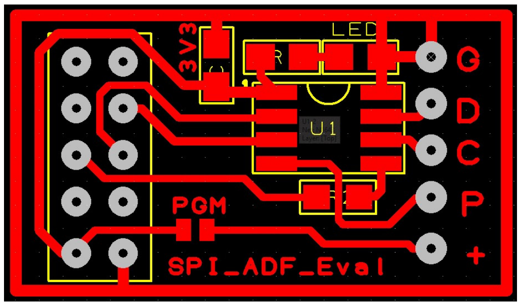


Photo 1 PIC + ADF5355 module



PCB Layout with 1:1 Mirrored artwork



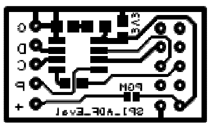
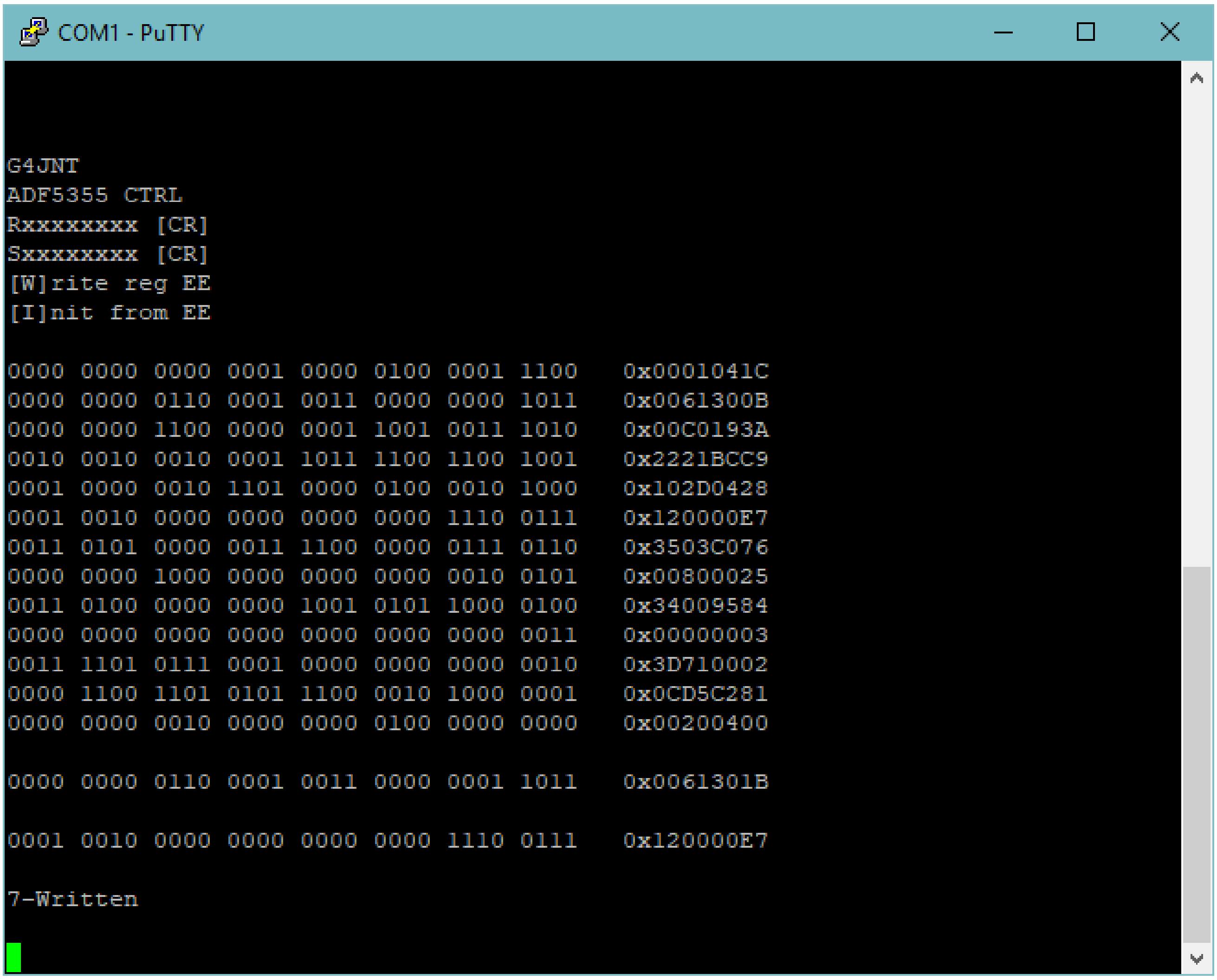


Figure 2 Screen Dump from Putty



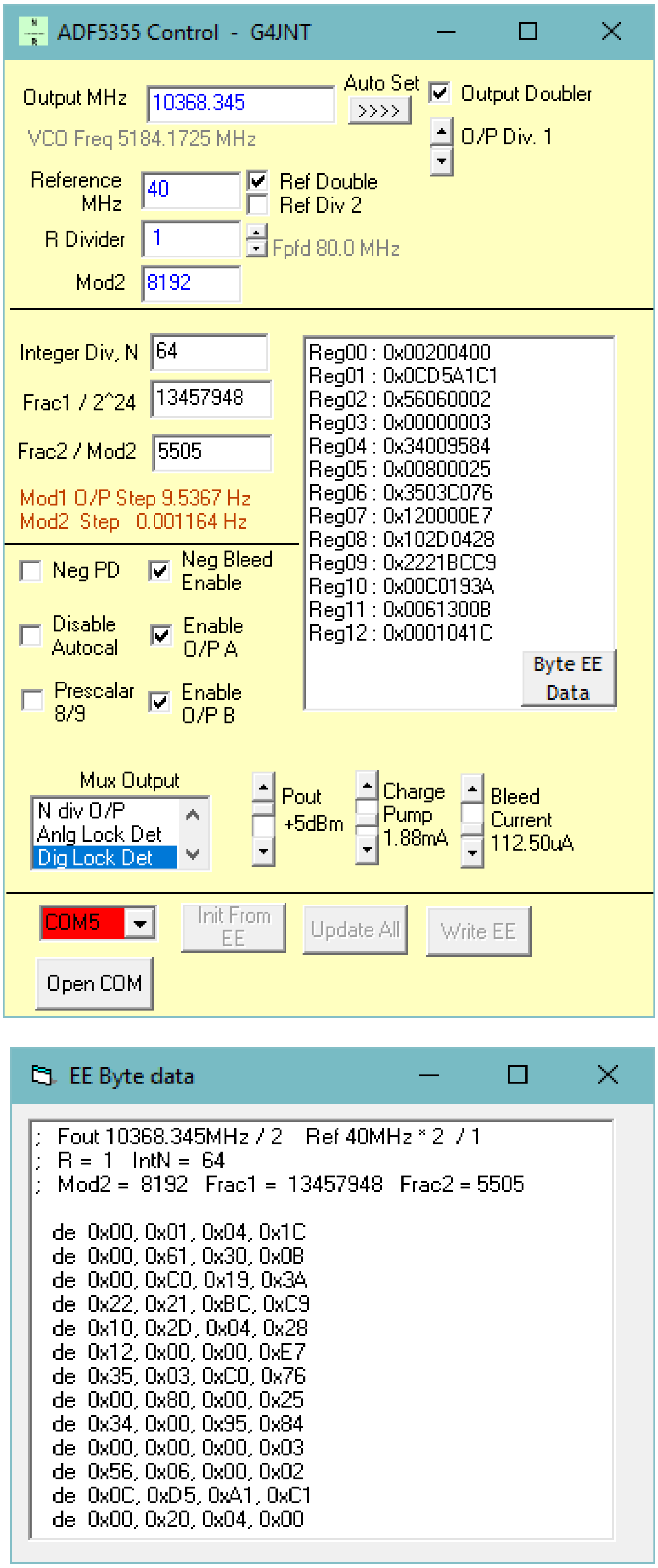
Control Software

For a complete understanding of how to use this synthesizer chip optimally, obtain a copy of the data sheet from http://www.analog.com/media/en/technical-documentation/data-sheets/ADF5355.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 ***ADF5355_Calc.EXE*** can be used in one of several ways. The main user screen can be seen in Figure 3. The user entries for frequency setting are into the windows at the top, with the figures in blue. The FINAL wanted output frequency is that after the output divider or doubler is applied. If the background to this window goes red then, depending on the frequency wanted, the doubler or output divider has to be changed to ensure the VCO operates in its correct range of 3.4 to 6.8GHz. The auto set button, labelled '>>>>' , can be used to automatically generate the correct settings here. The output frequency window will change to a white background when set properly .

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, Fpfd, as high as possible to reduce phase noise. The frequency doubler *RefDbl* can help here and, unless there is a very good reason to do otherwise, set R, the reference divider, to 1. *RefDiv2* is rarely needed – see the data sheet for why the option is there.

Figure 3 Screen Dump from ***ADF5355_CALC***



Mod2 is the second modulus for fine-setting. Its precise value for most practical purposes does not matter much – anything up to 16383 can be selected. 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 spurii. Consult the data sheet to see how to set these.

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

For communication with the PIC controller module use the selection box bottom left 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 '*Open COM*' 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 state of the output doubler cannot be determined from the read-back data, so if this is used the box must be checked <u>before</u> initialising from EE

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 '*Update All*' 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.

Closing the programme saves the entered values and some (but not all) of the check boxes to a file *ADF5355Ctl.ini* in the working directory. This is a plain text file that can be read and even edited if needed. The .INI file is recalled next time the software is run.

Calculating ADF5355 Register Values

The architectures used in the ADF5355 adopts a double modulus Fractional-N architecture that permits ultra-fine resolution to be achieved, but with a more complex programming process than earlier Fract-N devices. The output frequency is given by:

$F_{VCO} = F_{REF} / R' * (N + (F1 + F2 / D2) / 2^{24})$ and $F_{OUT} = F_{VCO} / Odiv$ or $2 * F_{VCO}$ at O/P B.

There are two modulus dividers, the first F1 works with a fixed modulus of 2^{24}, and the second F2 with the programmable D2. Rearranging the equation shows better how the ultrafine resolution is achieved. R' is the reference divider value, modified with *Ref Double* and/or *Ref Div 2* blocks.

$F_{VCO} = F_{REF} / R' * (N + F1 / 2^{24} + F2 / [D2 * 2^{24}])$

For a reference frequency of 40MHz, doubled to 80MHz, and an output from O/P B after the RF doubler of 10368.345MHz, the resulting F_{VCO} = 5184.175MHz

F_{VCO} / 80MHz = 64.80215625.

The Integer portion gives N = 64 to be put into the INT N register. Subtract this from the value calculated above, leaving a residue of 0.80215625.

Multiply the residue by 2^{24} , the fixed-value modulus of the first stage, to get the F1 value

$0.80215625 * 2^{24} = 13457948.672$

The integer portion of this is the F1 register, leaving a residue 0f 0.672

Choose a value for the second modulus D2. Maximum frequency resolution can be achieved by making this the highest allowed, 16383. Here we will [arbitrarily] use the nearest exact binary multiple below that, which is 8192. Multiply the final residue by D2 to get F2

0.672 * 8192 = 5505 discard any fractional part.

These register values can be seen in the screen dump.

$F_{VCO} = 80\text{MHz} * [64 + 13457948 / 2^{24} + 5505 / (8192 * 2^{24})]$

Ultimate frequency setting resolution is a function of F_{PFD} (the multiplied or divided F_{REF}) and the two modulus values. Resolution = $F_{PFD} / (2^{24} * D2)$ so for F_{PFD} = 80MHz and D2 = 8192, the VCO frequency step or resolution is around 0.00058Hz. This step is either doubled or divided by Odiv as appropriate.

When the device is used for real-time programming with frequencies calculated on the fly in a small microcontroller, it may be convenient to make D2 = 4096. Then the precise fractional frequency part is set by a continuous 36 bit word, the highest 24 bits forming F1 and the lower 12 bits F2. A value of 256 works even better , giving a 32 bit frequency word accessible as 3 + 1 bytes. Even here the resulting resolution will almost certainly be more than sufficient; 0.037Hz for the 10GHz example above.

All software with higher resolution copies of the diagrams can be found at

http://www.g4jnt.com/ADF5355_CTL.zip

UPDATE March 2020 **Rotary Encoder and Keypad Control Code now available**

http://g4jnt.com/ADF5355_Synth.pdf and http://g4jnt.com/adf5355_rot.zip

Registering MSCOMM32.OCX

If an error message appears that mentions this file when running ***ADF5355_Calc.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

Copy the supplied 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.