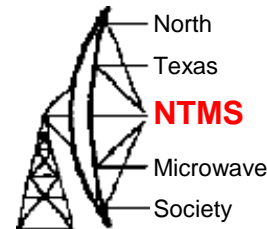
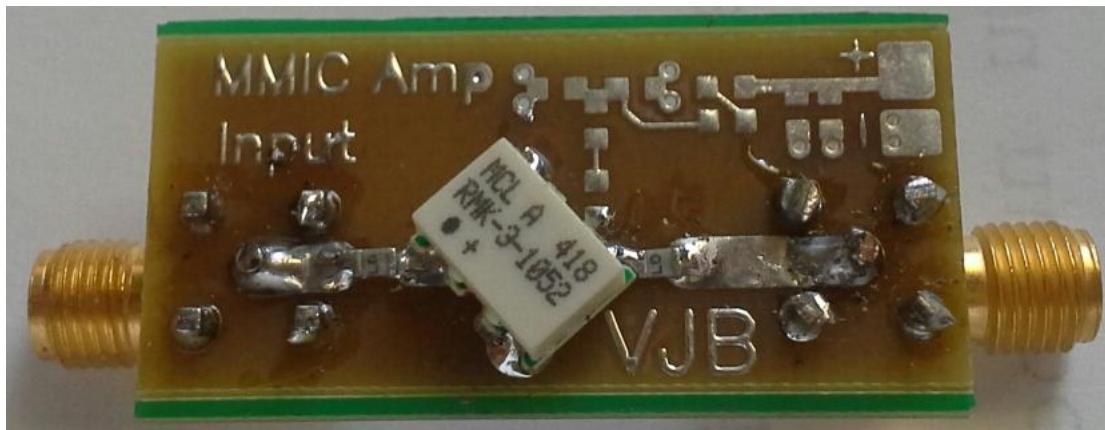


# Multipliers for 5616 MHz and 10224 MHz

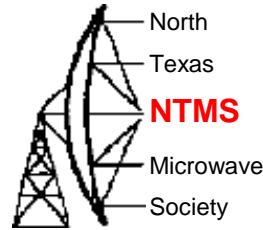


- MCL RMK-3-1052 Fits on WA5VJB MMIC Board



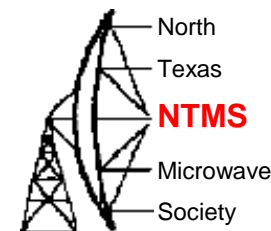
- KSX2-722 (X2) will take a new board
- Can buy ZX90-2-36S in package with connectors for \$37
- Mini-Circuits sell eval boards (board and two connectors) for each of the multipliers for about \$35 each
- Warning – do not orient package as shown (Pin 4 in input, not Pin 1)

# 432.380 MHz NTMS Beacon



- Merged and modified K6HX keyer code and F1CJN ADF4351 code
  - Using only the SPI bus interface from F1CJN code
    - Delays in human interface interfered with keyer timing
  - Hand calculated ADF4351 register values to use with 10 MHz and 25 MHz references (comment out unused set)
- Keyer output available on Pin A5.
  - Arduino SPI function uses the Arduino pin 13 which is also the on-board LED
  - Positive logic keyer output
- Programmed ADF4351 to accomplish RF on/off keying
  - On-the-air tests sound good
  - Pout = 1.5 dBm
- Added code for real-time temperature telemetry w/LM35

# ADF4351



## APPLICATIONS

Wireless infrastructure (W-CDMA, TD-SCDMA, WiMAX,  
GSM, PCS, DCS, DECT)

Test equipment

Wireless LANs, CATV equipment

Clock generation

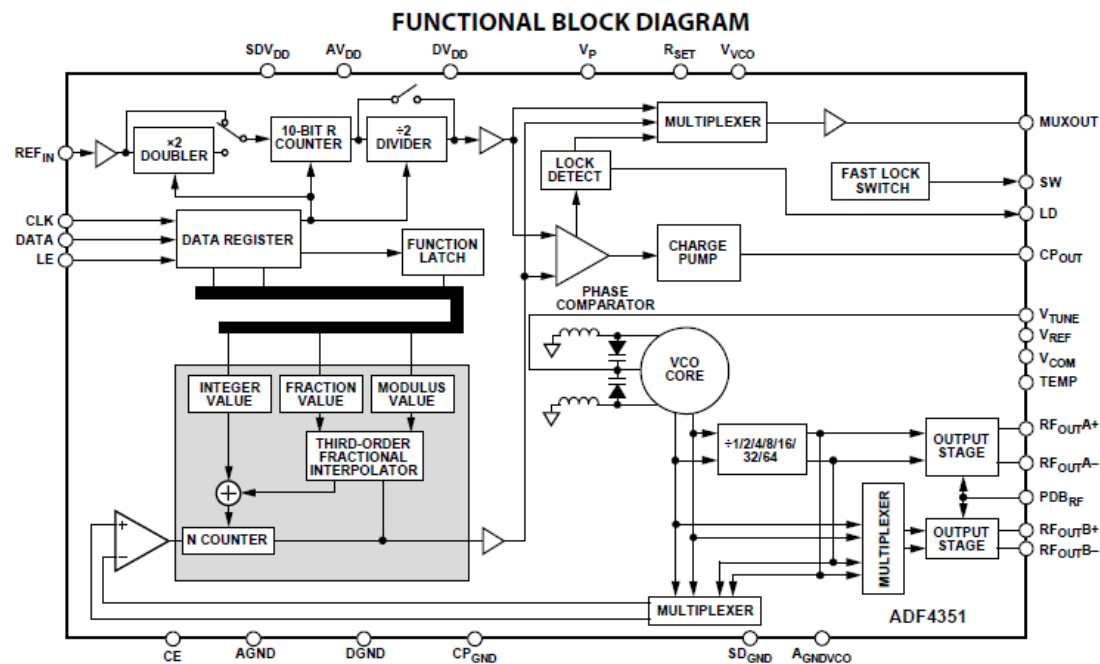


Figure 1.

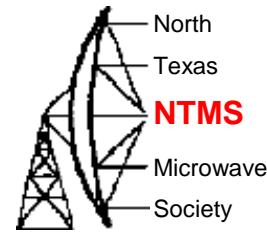
Rev. A

[Document Feedback](#)

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

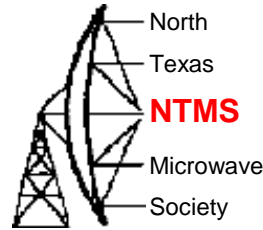
One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.  
Tel: 781.329.4700 ©2012–2017 Analog Devices, Inc. All rights reserved.  
[Technical Support](#) [www.analog.com](http://www.analog.com)

# Calculating Frequency Settings



- The RF VCO frequency (RFOUT) equation is  $RFOUT = fPFD \times (INT + (FRAC/MOD))$  where:
  - $RFOUT$  is the output frequency of the voltage controlled oscillator (VCO).
  - $INT$  is the preset divide ratio of the binary 16-bit counter (23 to 65,535 for the 4/5 prescaler; 75 to 65,535 for the 8/9 prescaler).
  - $FRAC$  is the numerator of the fractional division (0 to  $MOD - 1$ ).
  - $MOD$  is the preset fractional modulus (2 to 4095).

# Calculating Frequency Settings (2)



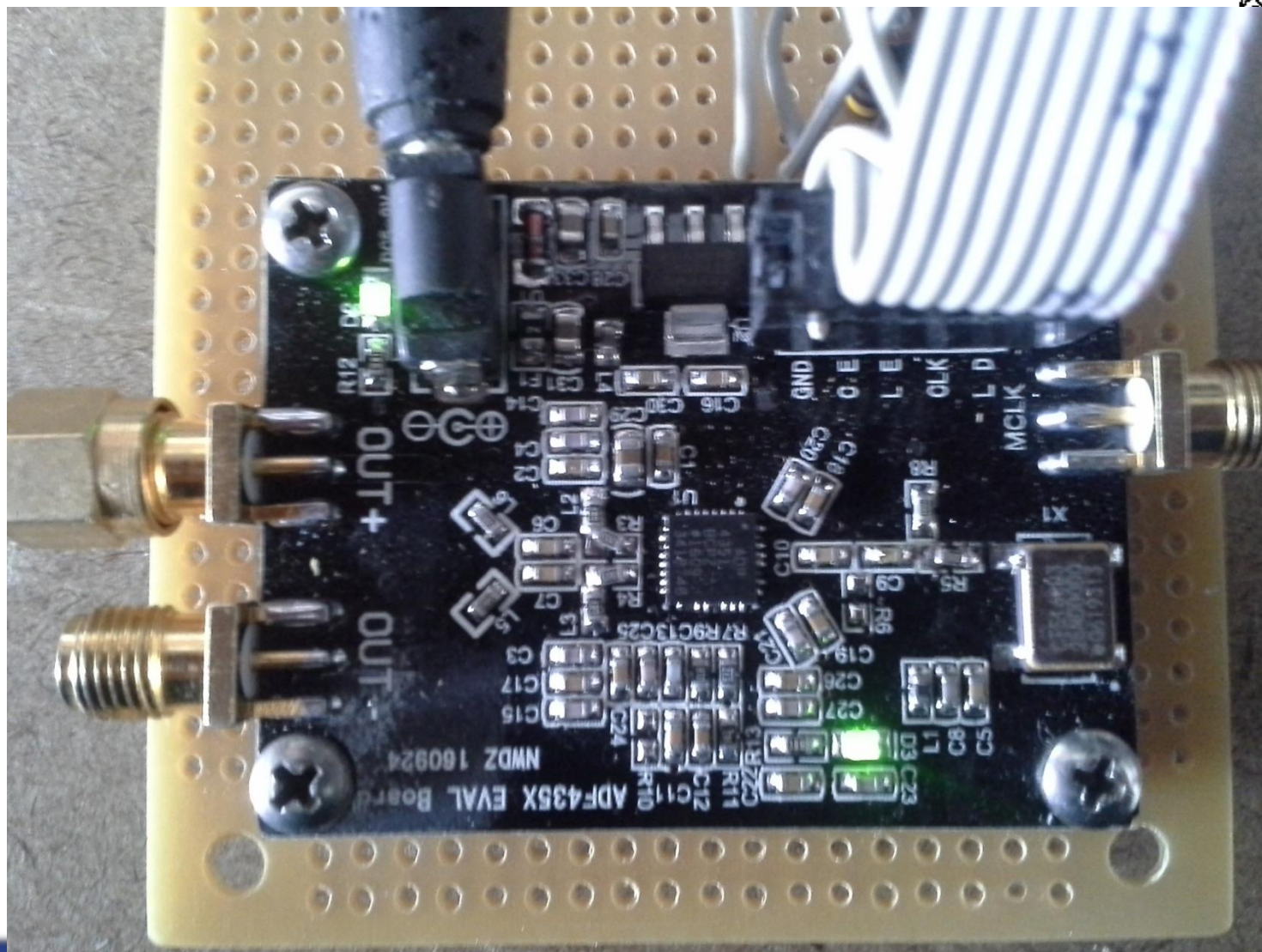
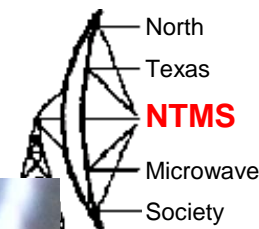
- *fPFD is the reference frequency*
  - $fPFD = REFIN \times [(1 + D)/(R \times (1 + T))]$  (2) where:
  - *REFIN* is the reference input frequency.
  - *D* is the REFIN doubler bit (0 or 1).
  - *R* is the preset divide ratio of the binary 10-bit programmable reference counter (1 to 1023).
  - *T* is the REFIN divide-by-2 bit (0 or 1).

# 432.380 MHz Example

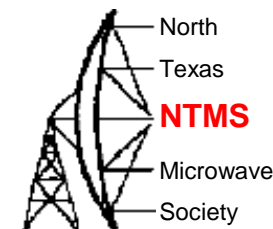
- Desired output = 432.380 MHz
- Reference = 10 MHz
- Frequency range required the VCO output to be divided by 8
- $432,380,000 = [\text{INT} + \text{FRAC}/\text{MOD}] * (10\text{MHz}/8)$ 
  - $\text{INT} + \text{FRAC}/\text{MOD} = 345.904$
  - $\text{INT} = 345$  or  $0x159$
  - $\text{FRAC}/\text{MOD} = 0.904$ 
    - Use  $\text{FRAC} = 904$  or  $0x388$
    - Used  $\text{MOD} = 1000$  or  $0x3E8$
- Map bits for INT and FRAC into Register 0
- Map bits for MOD into Register 1



# Chinese ADF4351 Eval Board



# Interface Between the Arduino Uno and the ADF3451 Eval Board



ADF4351 Signal	Arduino Function	Arduino Pin*	Direction	Level Shift Required
ADF Data	MOSI	11	To ADF4351	Yes
ADF Clock	SCK	13	To ADF4351	Yes
ADF LE	Select	3	To ADF4351	Yes
Muxout	Lock Detect	2	From ADF4351	No
+5VDC				
GND				

Interface via the LCD Button Shield and Interface Board

\*Per F1CJN software



# Arduino Uno Board

7 to 12VDC input  
2.1mm x 5.5mm  
Male center positive

Voltage regulator

16MHz crystal

ATmega16U2 microcontroller IC/USB controller

USB-B port to computer

Reset button

ICSP for USB interface

(I2C) SCL - Serial clock

(I2C) SDA - Serial data

Pin-13 LED

(SPI) SCK - Serial clock

(SPI) MISO - Master-in, slave-out

(SPI) MOSI - Master-out, slave-in

(SPI) SS - Slave select

Note: Pins denoted with "~" are PWM supported

Interrupt 1

Interrupt 2

TXD

RXD

VCC

MOSI

GND

RESET

SCK

MISO

ICSP for ATmega328

ATmega328 microcontroller IC

(I2C) SDA

(I2C) SCL

Analog pin 0

Analog pin 1

Analog pin 2

Analog pin 3

Input voltage

Ground

5V Output

3.3V Output

Reset

I/O Reference voltage

Not connected

IOREF

RESET

3.3V

5V

GND

GND

Vin

A0

A1

A2

A3

A4

A5

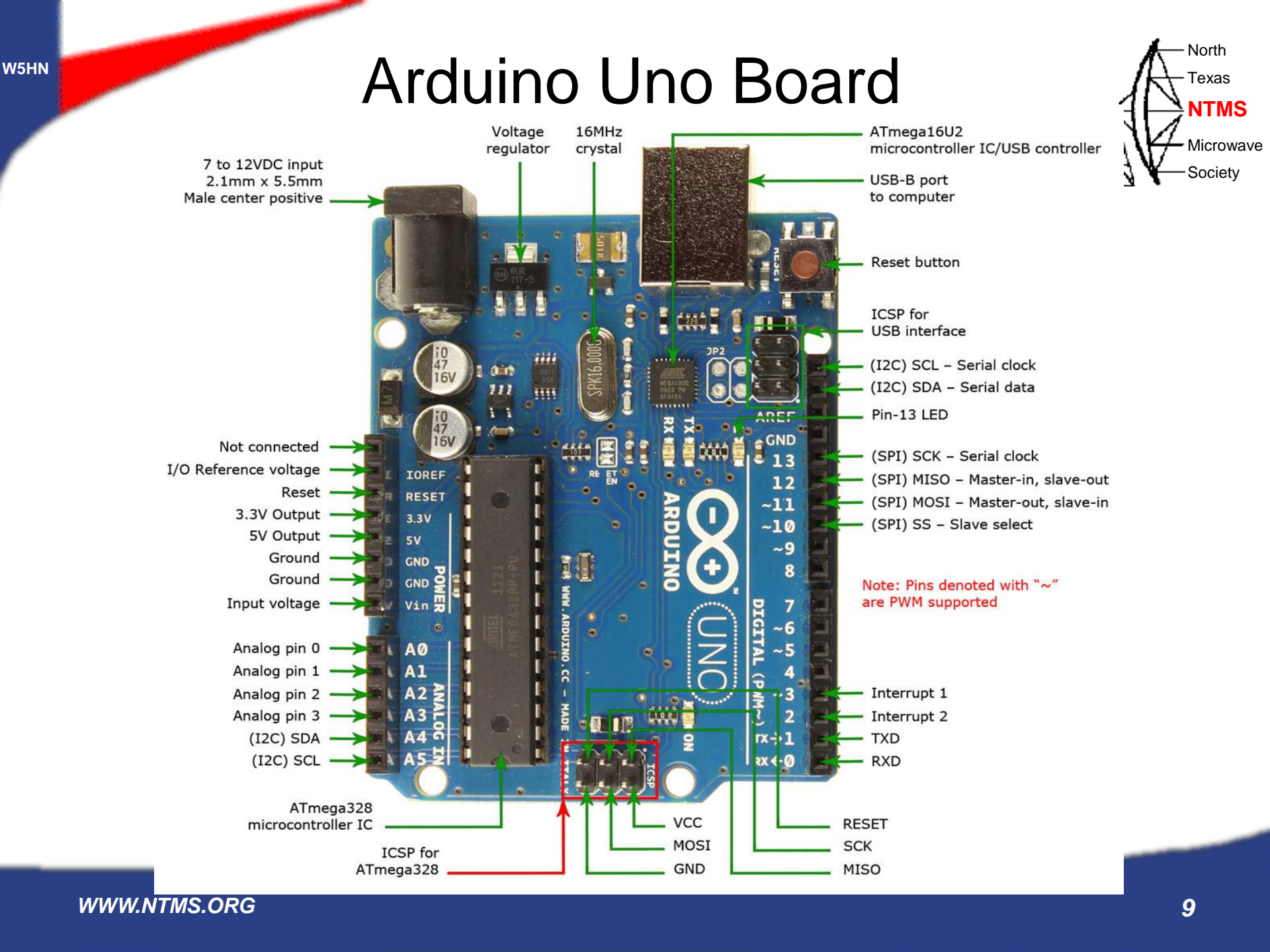
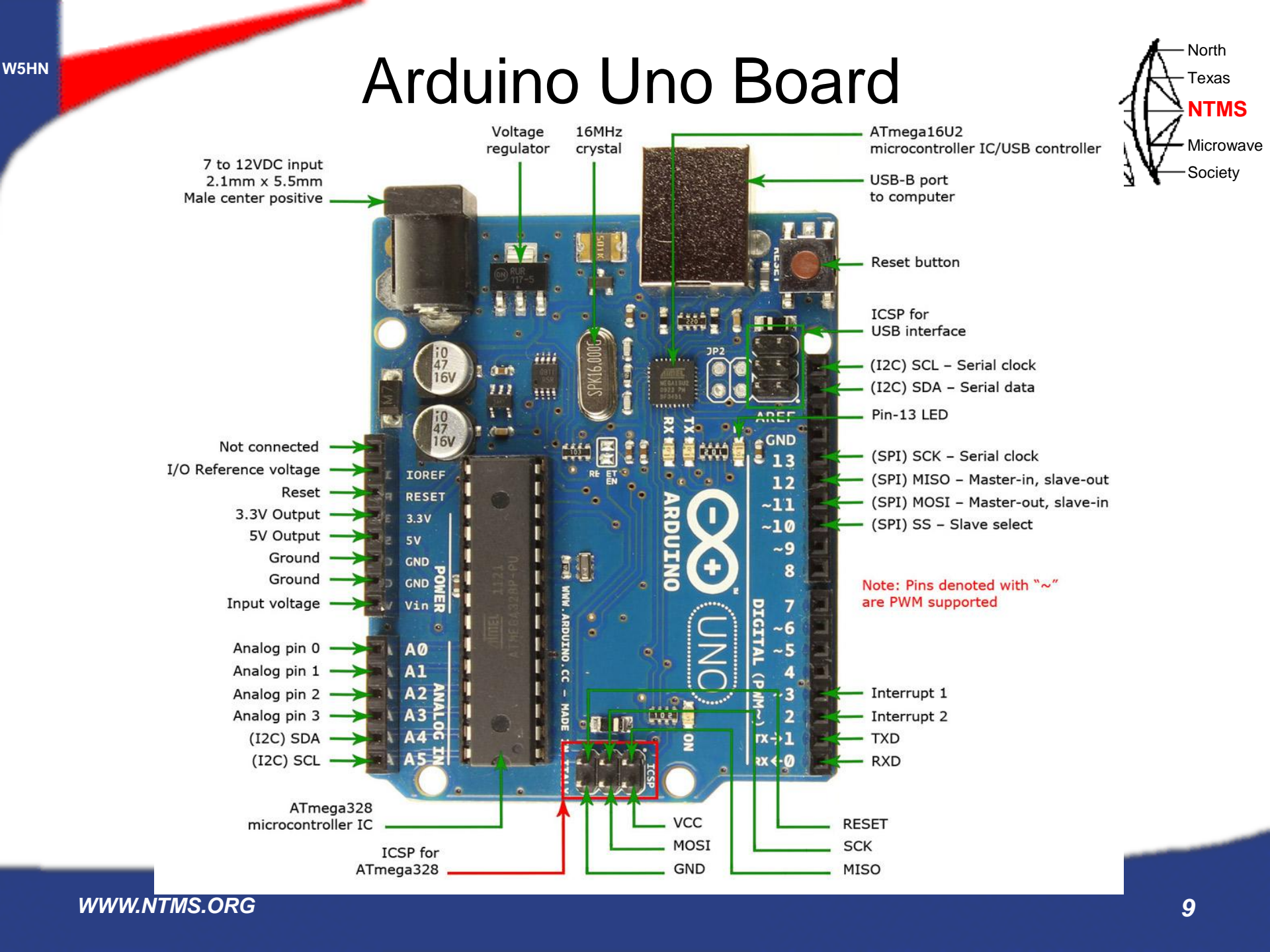
ARDUINO

UNO

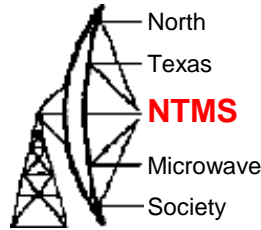
W55HN

WWW.NTMS.ORG

9

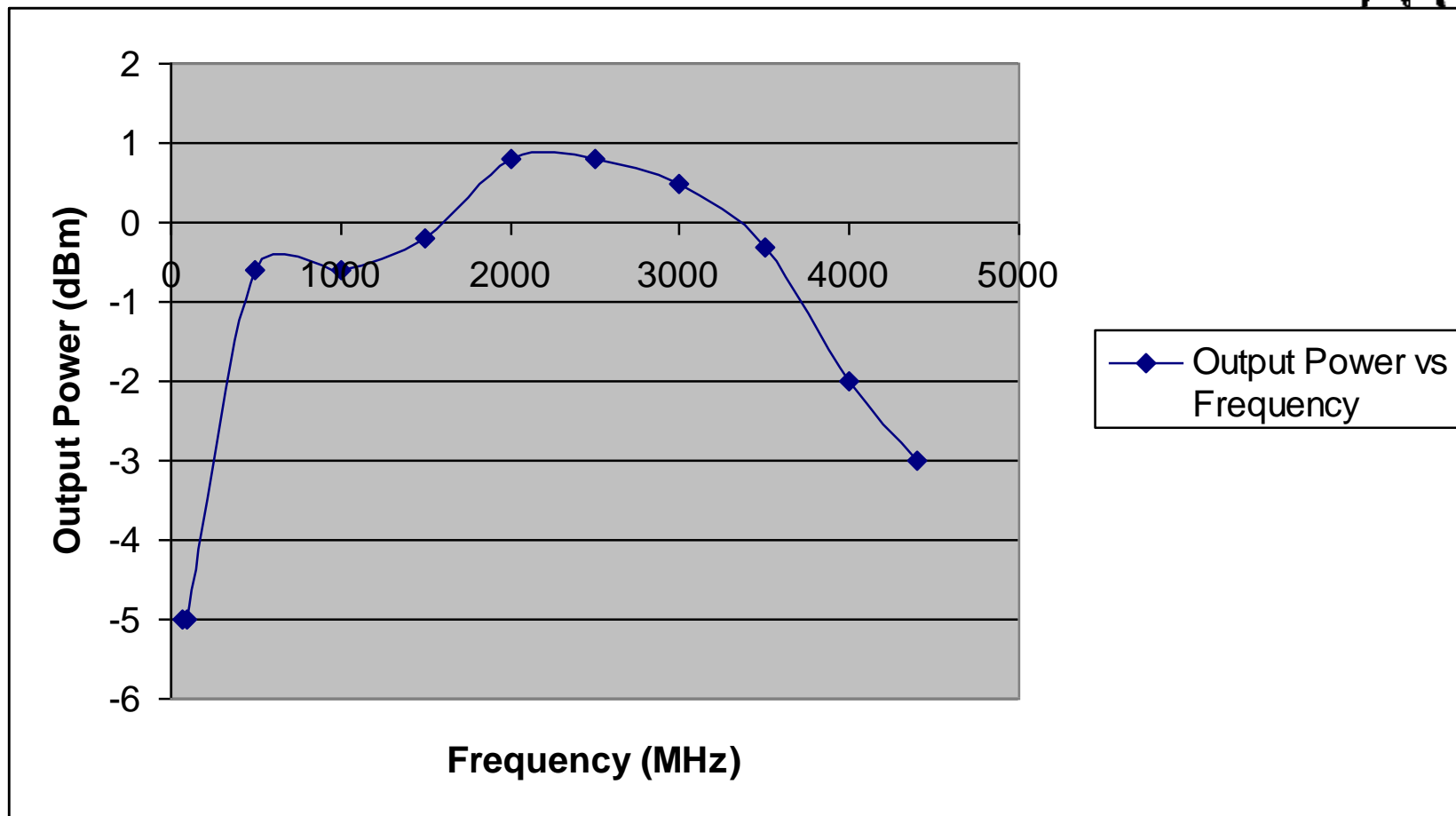
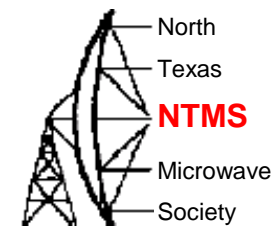


# F1CJN Software



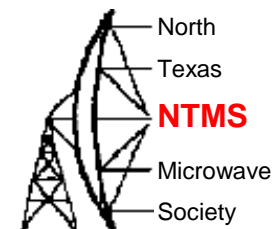
- Alain Fort, F1CJN wrote software for controlling the ADF4351
- Code compiles and loads on the Arduino Uno and available at
  - [http://f6kbf.free.fr/html/ADF4351%20and%20Arduino\\_Fr\\_Gb.htm](http://f6kbf.free.fr/html/ADF4351%20and%20Arduino_Fr_Gb.htm) (thanks to WW2R and W5LUA)

# Measured Output Power\*

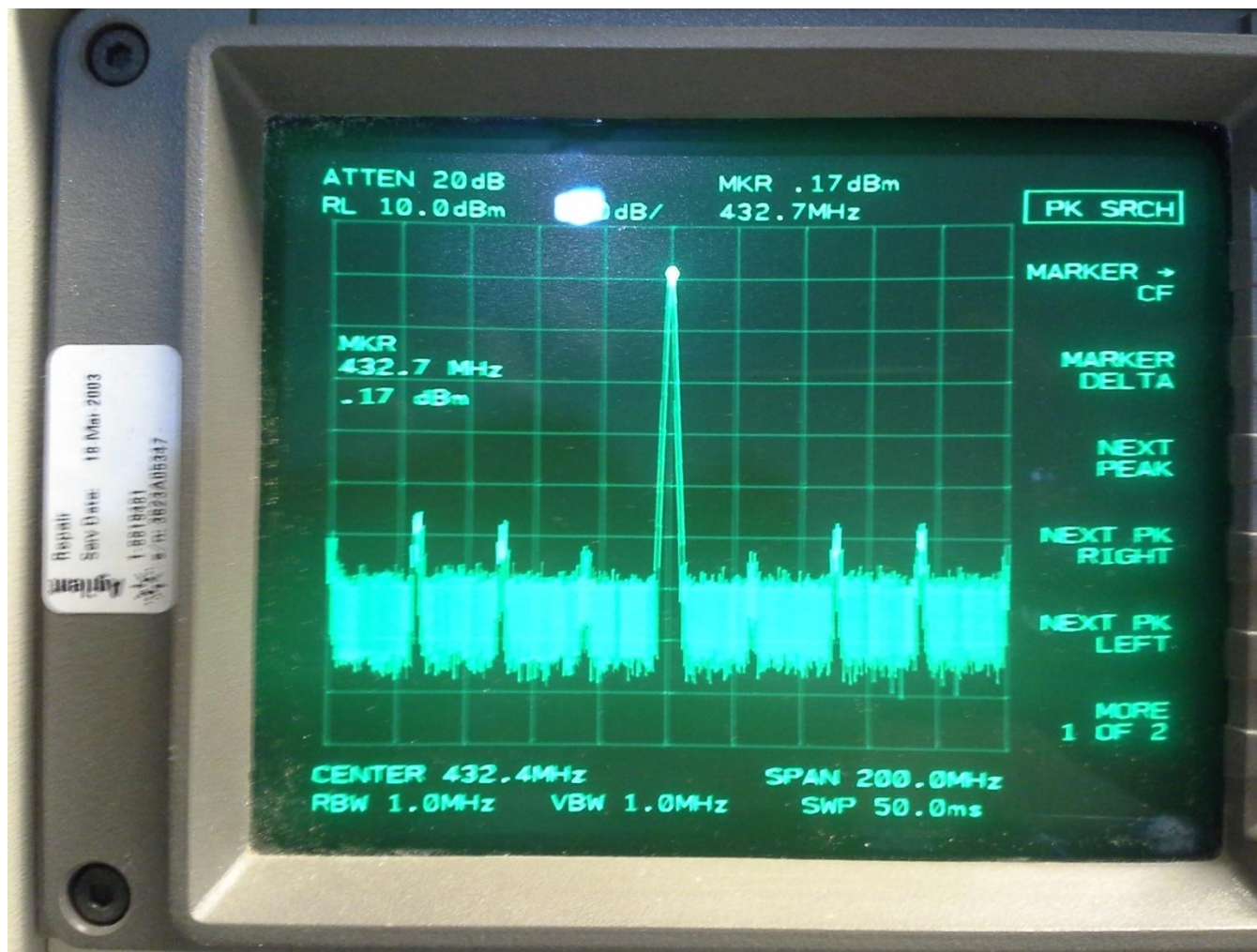


\*Single ended with other polarity terminated

# Spectrum: 432.280 MHz

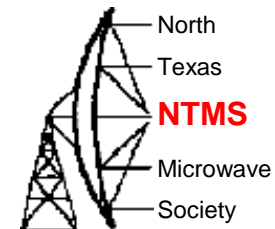


With on-  
board 25  
MHz  
XTAL

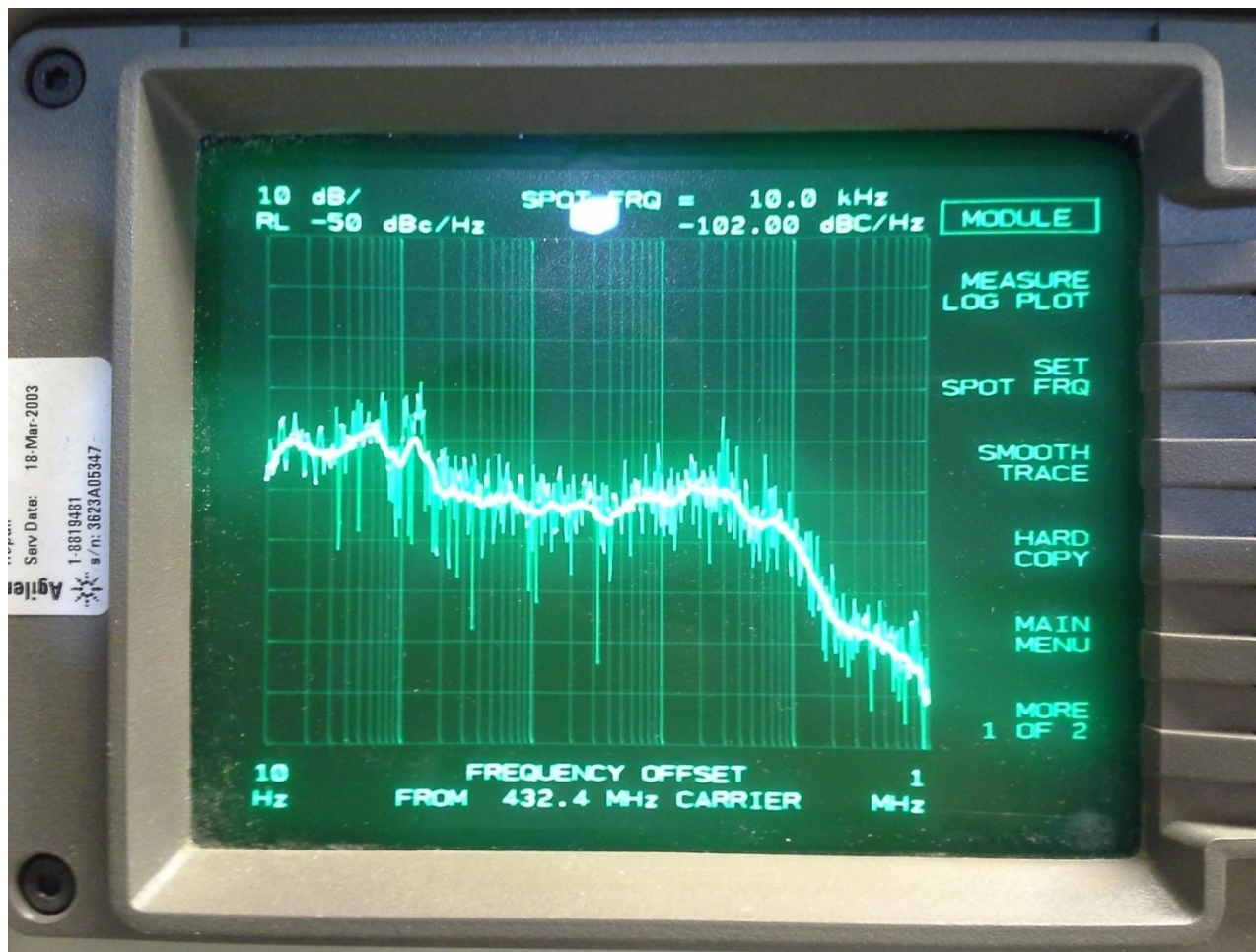


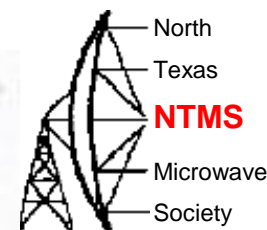
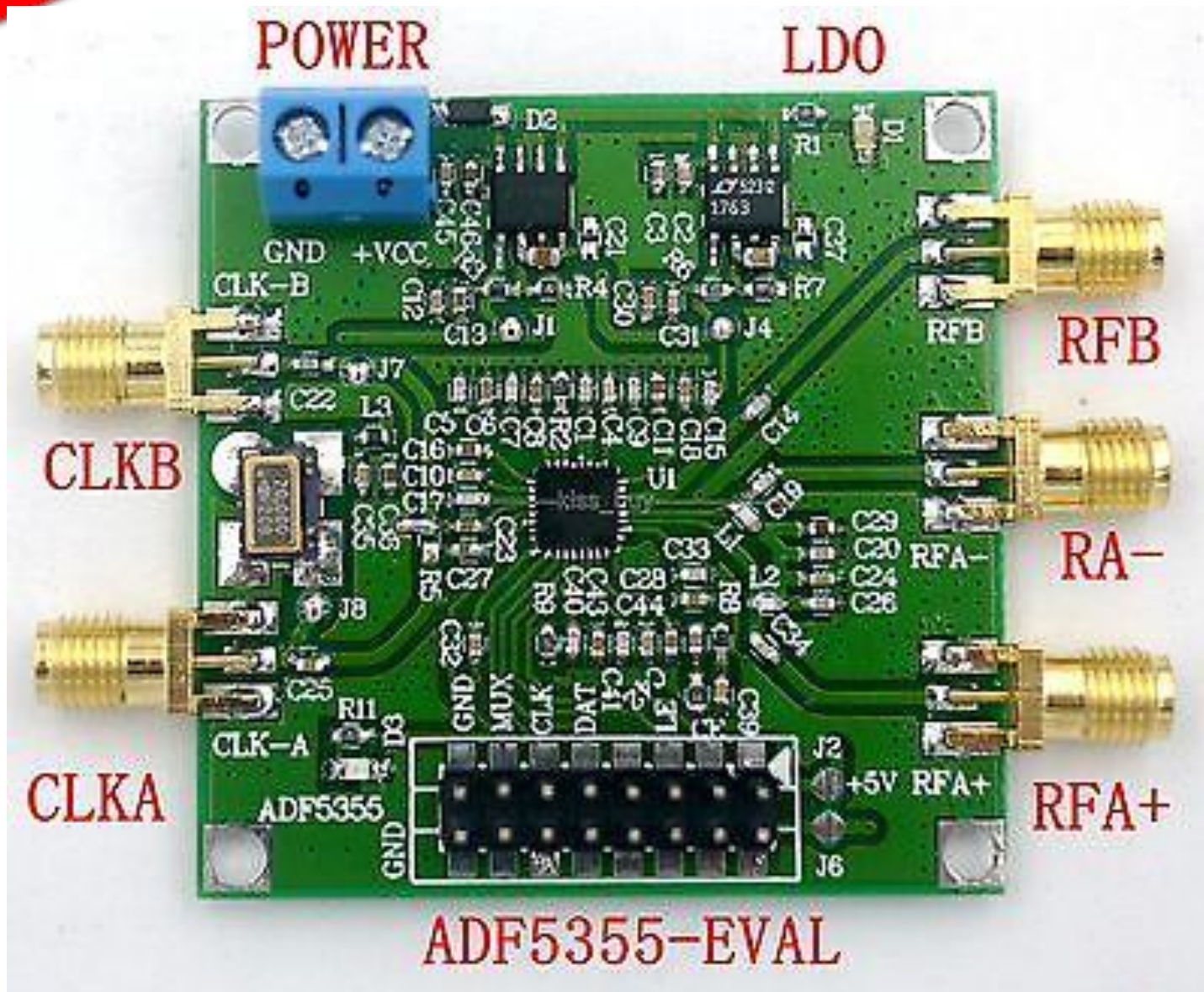


# Phase Noise: 432.380 MHz



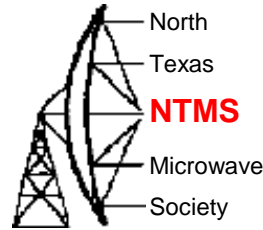
With on-board 25 MHz XTAL







# ADF5355 Eval Board



- On-board XTAL reference is 20 MHz
- Separate supplies for logic (3.3V) and RF (5V)
- Differential external reference input – can use single-ended
- Two separate RF Outputs
  - RF Output A
    - Differential
    - 54 MHz to 6.8 GHz
  - RF Output B
    - Single-ended
    - 6.8 GHz to 13.6 GHz
- Hand calculated register values for 5616 MHz and 10224 MHz with 10 MHz and 20 MHz references
- Wrote control program for the ADF5355 using F1CJN's interface to the LCD Button Shield

# PLANS

- Build up interface for ADF5355 eval board and test with:
  - Fixed, calculated register values
  - Code for user entered frequency values
- Check ADF5355 performance
- Continue work with ADF4351
  - Frequency multipliers
- No further work on retaining register values on power off
  - Can be accomplished with a backup battery
  - Easier to spend the few extra \$ for an Arduino