

Summary of information to interface the MSP430 with the ADS1298. Info taken from TI Documentation and gathered from E2E forums hosted by TI. Author: Jesus Efrain Gaxiola-Sosa, efraingaxiola@gmail.com. **Disclaimer: The rights for the figures and text definitions showed on this document belong to Texas Instruments® and are used for education purposes only.**

## Understanding how to use the ADS1298

The ADS1294/6/8/4R/6R/8R are a family of multichannel, simultaneous sampling, 24-bit, **delta-sigma (  $\Delta \Sigma$  ) analog-to-digital converters (ADCs) with built-in programmable gain amplifiers (PGAs)**, internal reference, and an onboard oscillator.

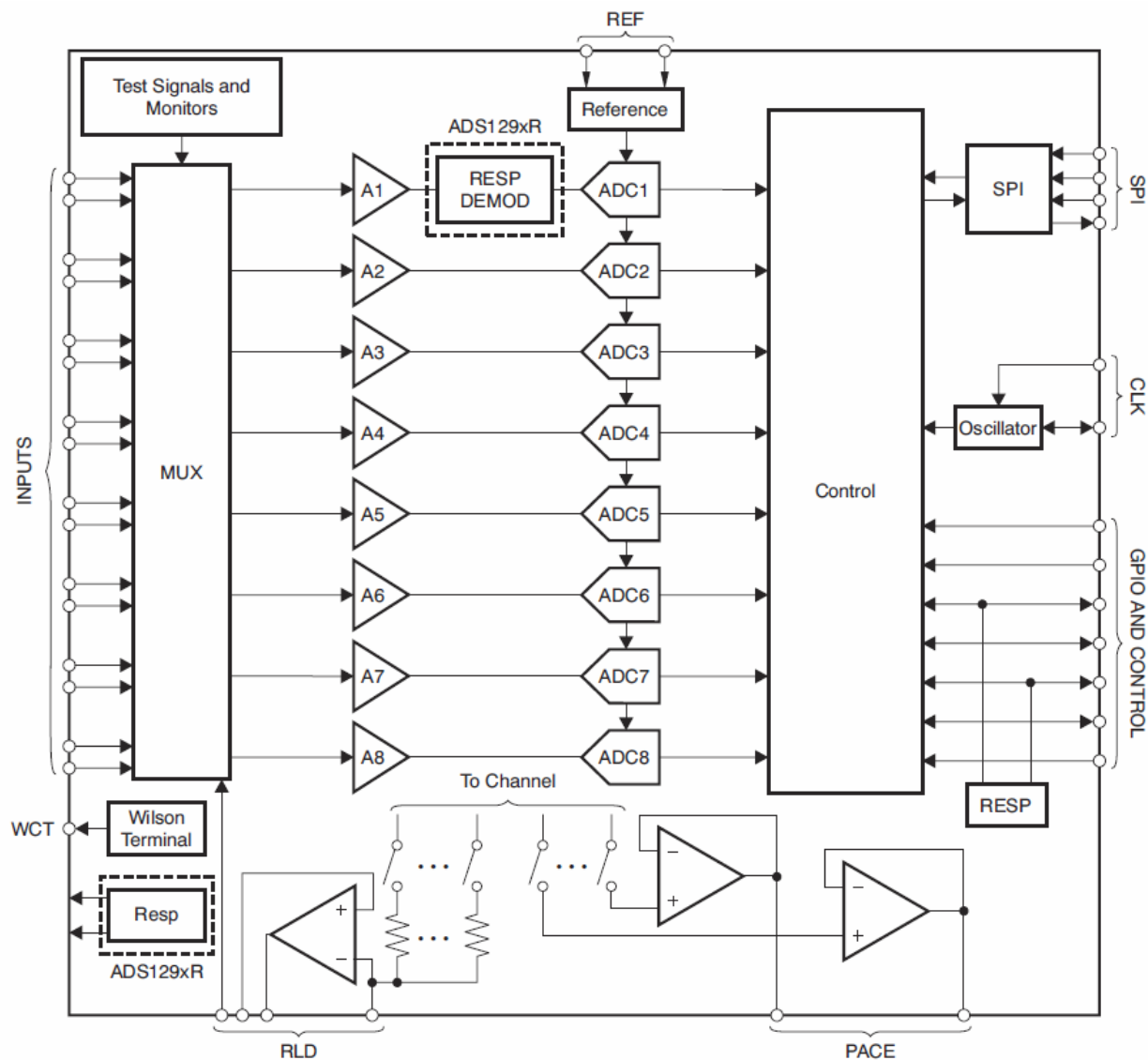
Any configuration of input channels can be selected for derivation of the right leg drive (RLD) output signal.

Three integrated amplifiers generate the Wilson Central Terminal (WCT) and the Goldberger Central Terminals (GCT) required for a standard 12-lead ECG.

The ADCs in the device offer data rates from 250SPS to 32kSPS. Communication to the device is accomplished using an SPI-compatible interface. The device provides four GPIO pins for general use.

Monitoring techniques (such as lead-off detection) verify that electrodes are properly connected, and immediately notify the user if a fault is detected. This fault can be configured to alarm when an Electrode is completely disconnected or when the connection is weak. Lead-off detection can be accomplished either by using a pull-up/pull-down resistor or a current source/sink. An internal ac lead-off detection feature is also available.

Summary of information to interface the MSP430 with the ADS1298. Info taken from TI Documentation and gathered from E2E forums hosted by TI. Author: Jesus Efrain Gaxiola-Sosa, efraingaxiola@gmail.com. **Disclaimer: The rights for the figures and text definitions showed on this document belong to Texas Instruments® and are used for education purposes only.**

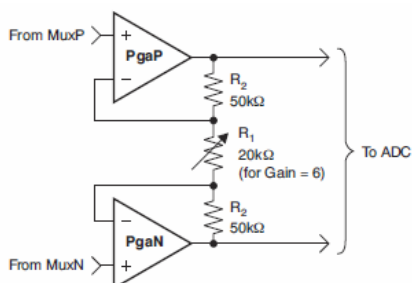




Summary of information to interface the MSP430 with the ADS1298. Info taken from TI Documentation and gathered from E2E forums hosted by TI. Author: Jesus Efrain Gaxiola-Sosa, efraingaxiola@gmail.com. **Disclaimer: The rights for the figures and text definitions showed on this document belong to Texas Instruments® and are used for education purposes only.**

## PGA

The PGA is a differential input/differential output amplifier, as shown in Figure 28. It has seven gain settings (1, 2, 3, 4, 6, 8, and 12) that can be set by writing to the CHnSET register.



**Figure 28. PGA Implementation**

**Table 6. PGA Gain versus Small-Signal Bandwidth**

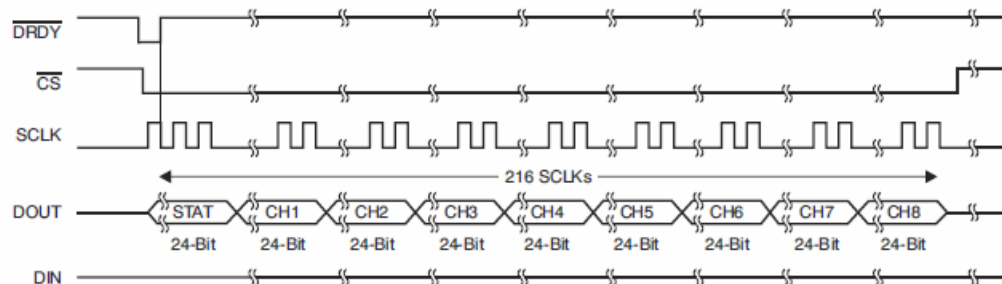
GAIN	NOMINAL BANDWIDTH AT ROOM TEMPERATURE (kHz)
1	237
2	146
3	127
4	96
6	64
8	48
12	32

## DATA FORMAT

The ADS129x output 24 bits of data per channel in binary two's complement format, MSB first. The LSB has a weight of  $V_{REF}/(223 - 1)$ . A positive full-scale input produces an output code of 7FFFFh and the negative full-scale input produces an output code of 80000h. The output clips at these codes for signals exceeding full-scale.

## SPI INTERFACE

The SPI-compatible serial interface consists of four signals: CS, SCLK, DIN, and DOUT. The interface reads conversion data, reads and writes registers, and controls the ADS129x operation. The DRDY output is used as a status signal to indicate when data are ready. DRDY goes low when new data are available.



**Figure 37. SPI Bus Data Output for the ADS1298 (Eight Channels)**

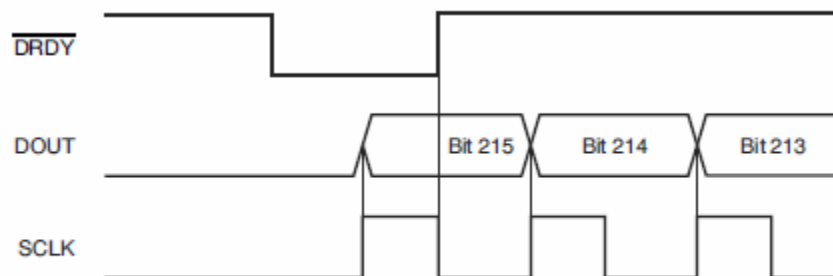
## Data Retrieval

Data retrieval can be accomplished in one of two methods. The read data continuous command (see the *RDATA\_C: Read Data Continuous* section) can be used to set the device in a mode to read the data continuously without sending opcodes. The read data command (see the *RDATA: Read Data* section) can be used to read just one data output from the device (see the *SPI Command Definitions* section for more details). The conversion data are read by shifting the data out on DOUT. The MSB of the data on DOUT is clocked out on the first SCLK rising edge. DRDY returns to high on the first SCLK falling edge. DIN should remain low for the entire read operation.

The number of bits in the data output depends on the number of channels and the number of bits per channel. **For the ADS1298/8R, the number of data outputs is (24 status bits + 24 bits × 8 channels) = 216 bits. The format of the 24 status bits is: (1100 + LOFF\_STATP + LOFF\_STATN + bits[4:7] of the GPIO register). The data format for each channel data are twos complement and MSB first.** When channels are powered down using the user register setting, the corresponding channel output is set to '0'. However, the sequence of channel outputs remains the same. For the ADS1294/4R and ADS1296/6R, the last four and two channel outputs shown in Figure 37 are zeros. The four and six channels parts require only 120 and 168 SCLKs to shift data out, respectively. Status and GPIO register bits are loaded into the 24-bit status word 2tCLKs before DRDY goes low. The ADS129x also provide a multiple readback feature. The data can be read out multiple times by simply giving more SCLKs, in which case the MSB data byte repeats after reading the last byte. The DAISY\_EN bit in CONFIG1 register must be set to '1' for multiple readbacks.

## Data Ready (DRDY)

DRDY is an output. When it transitions low, new conversion data are ready. Figure 38 shows the relationship between DRDY, DOUT, and SCLK during data retrieval (in case of an ADS1298 with a selected data rate that gives 24-bit resolution). DOUT is latched out at the rising edge of SCLK. DRDY is pulled high at the falling edge of SCLK. Note that DRDY goes high on the first falling edge SCLK regardless of whether data are being retrieved from the device or a command is being sent through the DIN pin.



**Figure 38.  $\overline{\text{DRDY}}$  with Data Retrieval ( $\overline{\text{CS}} = 0$  in RDATA mode)**

## Continuous Mode

Conversions begin when the START pin is taken high for at least 2 tCLKs or when the START opcode command is sent. As seen in Figure 41, the DRDY output goes high when conversions are started and

Summary of information to interface the MSP430 with the ADS1298. Info taken from TI Documentation and gathered from E2E forums hosted by TI. Author: Jesus Efrain Gaxiola-Sosa, efraingaxiola@gmail.com. **Disclaimer: The rights for the figures and text definitions showed on this document belong to Texas Instruments® and are used for education purposes only.**

goes low when data are ready. Conversions continue indefinitely until the START pin is taken low or the STOP opcode command is transmitted. When the START pin is pulled low or the stop command is issued, the conversion in progress is allowed to complete. Figure 42 and Table 10 show the required timing of DRDY to the START pin and the START/STOP opcode commands when controlling conversions in this mode. To keep the converter running continuously, the START pin can be permanently tied high. Note that when switching from pulse mode to continuous mode, the START signal is pulsed or a STOP command must be issued followed by a START command. This conversion mode is ideal for applications that require a fixed-continuous stream of conversions results.

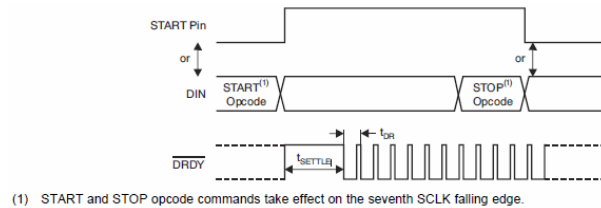


Figure 41. Continuous Conversion Mode

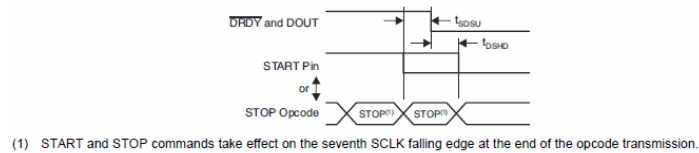


Figure 42. START to DRDY Timing

## WILSON CENTRAL TERMINAL (WCT) AND CHEST LEADS

In the standard 12-lead ECG, WCT voltage is defined as the average of Right Arm (RA), Left Arm (LA), and Left Leg (LL) electrodes. This voltage is used as the reference voltage for the measurement of the chest leads. The ADS129x has three integrated low-noise amplifiers that generate the WCT voltage. Figure 53 shows the block diagram of the implementation.

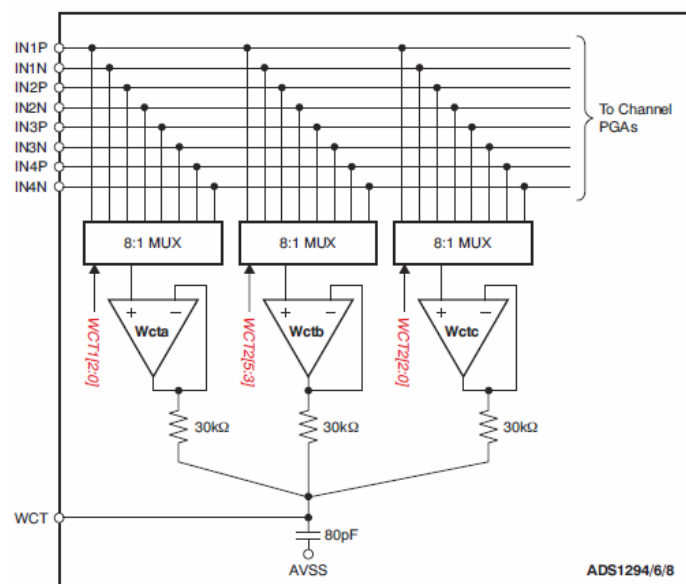


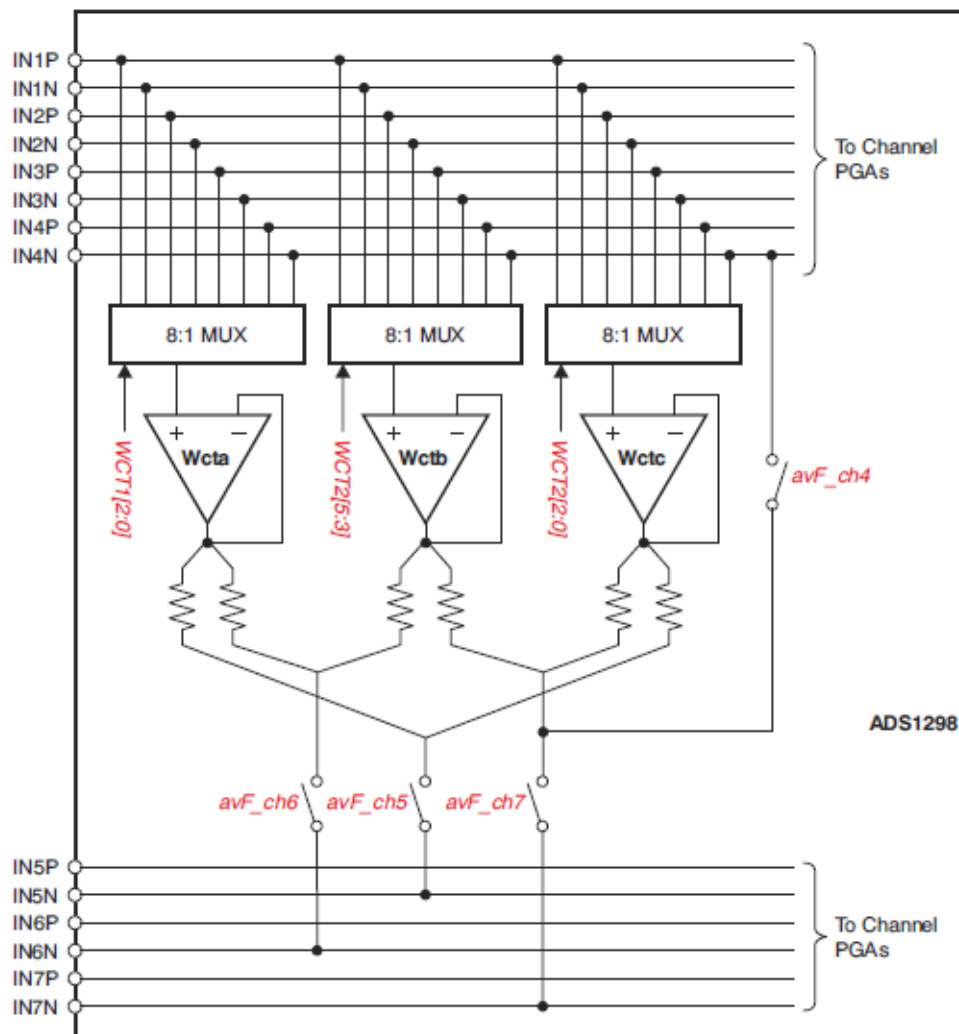
Figure 53. WCT Voltage

Summary of information to interface the MSP430 with the ADS1298. Info taken from TI Documentation and gathered from E2E forums hosted by TI. Author: Jesus Efrain Gaxiola-Sosa, efraingaxiola@gmail.com. **Disclaimer: The rights for the figures and text definitions showed on this document belong to Texas Instruments® and are used for education purposes only.**

The devices provide flexibility to choose any one of the eight signals (IN1P to IN4N) to be routed to each of the amplifiers to generate the average. Having this flexibility allows the RA, LA, and LL electrodes to be connected to any input of the first four channels depending on the lead configuration.

### Augmented Leads

In the typical implementation of the 12-lead ECG with eight channels, the augmented leads are calculated digitally. In certain applications, it may be required that all leads be derived in analog rather than digital. The ADS1298/8R provides the option to generate the augmented leads by routing appropriate averages to channels 5 to 7. **The same three amplifiers that are used to generate the WCT signal are used to generate the Goldberger Central Terminal signals as well.** Figure 56 shows an example of generating the augmented leads in analog domain. **Note that in this implementation it takes more than eight channels to generate the standard 12 leads.** Also, this feature is not available in the ADS1296/6R and ADS1294/4R.



**Figure 56. Analog Domain Augmented Leads**

## Right Leg Drive with the WCT Point

In certain applications, the out-of-phase version of the WCT is used as the right leg drive reference. The ADS1298 provides the option to have a buffered version of the WCT terminal at the RLD\_OUT pin. This signal can be inverted in phase using an external amplifier and used as the right leg drive. Refer to the *Right Leg Drive (RLD DC Bias Circuit)* section for more details.

### WCT as RLD

In certain applications, the right leg drive is derived as the average of RA, LA, and LL. This level is the same as the WCT voltage. The WCT amplifier has limited drive strength and thus should be used only to drive very high impedances directly. The ADS129x provide an option to internally buffer the WCT signal by setting the WCT\_TO\_RLD bit in the CONFIG4 register. The RLD\_OUT and RLD\_INV pins should be shorted external to the device. Note that before the RLD\_OUT signal is connected to the RLD electrode, an external amplifier should be used to invert the phase of the signal for negative feedback.

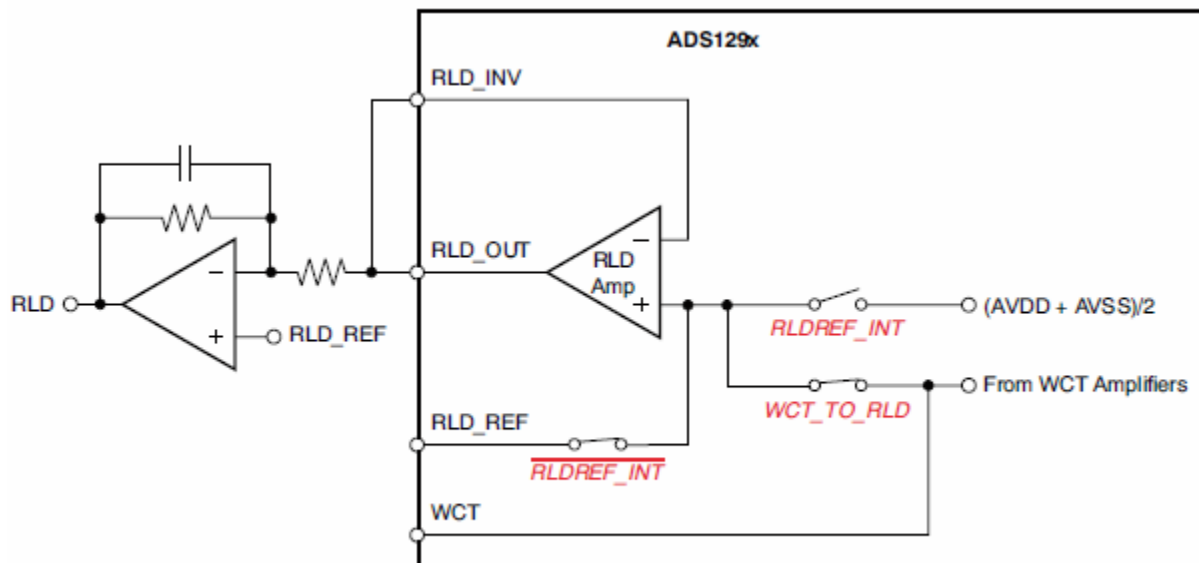


Figure 62. Using the WCT as the Right Leg Drive

## PCB LAYOUT

### Power Supplies and Grounding

The ADS129x have three supplies: AVDD, AVDD1, and DVDD. Both AVDD and AVDD1 should be as quiet as possible. AVDD1 provides the supply to the charge pump block and has transients at fCLK. Therefore, it is recommended that AVDD1 and AVSS1 be star-connected to AVDD and AVSS. It is important to eliminate noise from AVDD and AVDD1 that is non-synchronous with the ADS129x operation. Each supply of the ADS129x should be bypassed with 1  $\mu$ F and 0.1  $\mu$ F solid ceramic capacitors. It is recommended that placement of the digital circuits (DSP, microcontrollers, FPGAs, etc.) in the system is done such that the return currents on those devices do not cross the analog return path of the ADS129x.



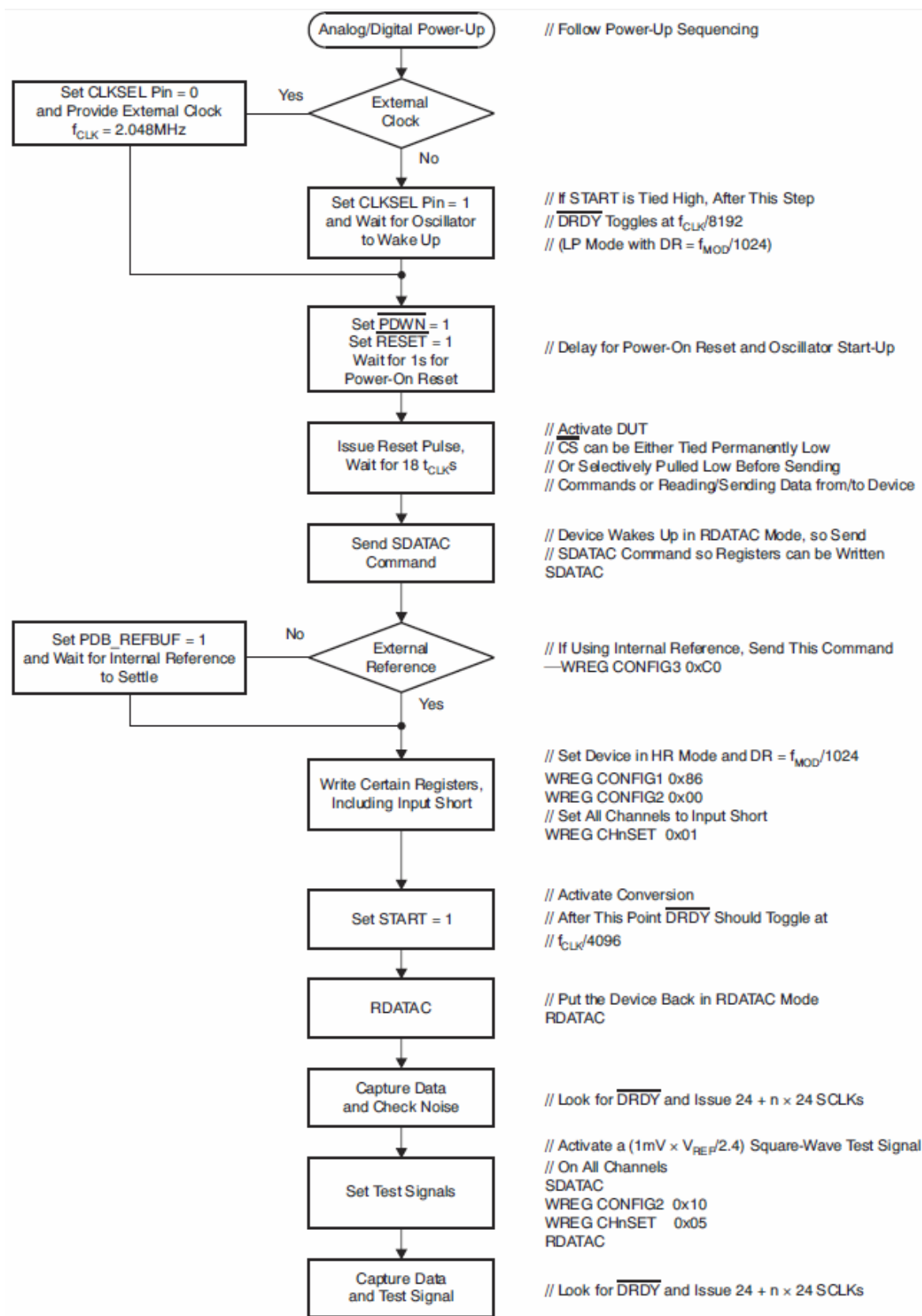
Summary of information to interface the MSP430 with the ADS1298. Info taken from TI Documentation and gathered from E2E forums hosted by TI. Author: Jesus Efrain Gaxiola-Sosa, efraingaxiola@gmail.com. **Disclaimer: The rights for the figures and text definitions showed on this document belong to Texas Instruments® and are used for education purposes only.**

The ADS129x can be powered from unipolar or bipolar supplies. Capacitors used for decoupling can be of surface-mount, low-cost, low-profile, multi-layer ceramic type. In most cases, the VCAP1 capacitor can also be a multi-layer ceramic, but in systems where the board is subjected to high- or low-frequency vibration, it is recommend to install a non-ferroelectric capacitor such as a tantalum or class 1 capacitor (COG or NPO). EIA class 2 and class 3 dielectrics such as (X7R, X5R, X8R, etc.) are ferroelectric. The piezoelectric property of these capacitors can appear as electrical noise coming from the capacitor. When using internal reference, noise on the VCAP1 node results in performance degradation.

## **SETTING THE DEVICE FOR BASIC DATA CAPTURE**

The following section outlines the procedure to configure the device in a basic state and capture data. This procedure is intended to put the device in a data sheet condition to check if the device is working properly in the user's system. It is recommended that this procedure be followed initially to get familiar with the device settings. Once this procedure has been verified, the device can be configured as needed. For details on the timings for commands refer to the appropriate sections in the data sheet. Also, some sample programming codes are added for the ECG-specific functions.

Summary of information to interface the MSP430 with the ADS1298. Info taken from TI Documentation and gathered from E2E forums hosted by TI. Author: Jesus Efrain Gaxiola-Sosa, efraingaxiola@gmail.com. **Disclaimer: The rights for the figures and text definitions showed on this document belong to Texas Instruments® and are used for education purposes only.**



**Figure 77. Initial Flow at Power-Up**

Summary of information to interface the MSP430 with the ADS1298. Info taken from TI Documentation and gathered from E2E forums hosted by TI. Author: Jesus Efrain Gaxiola-Sosa, efraingaxiola@gmail.com. **Disclaimer: The rights for the figures and text definitions showed on this document belong to Texas Instruments® and are used for education purposes only.**

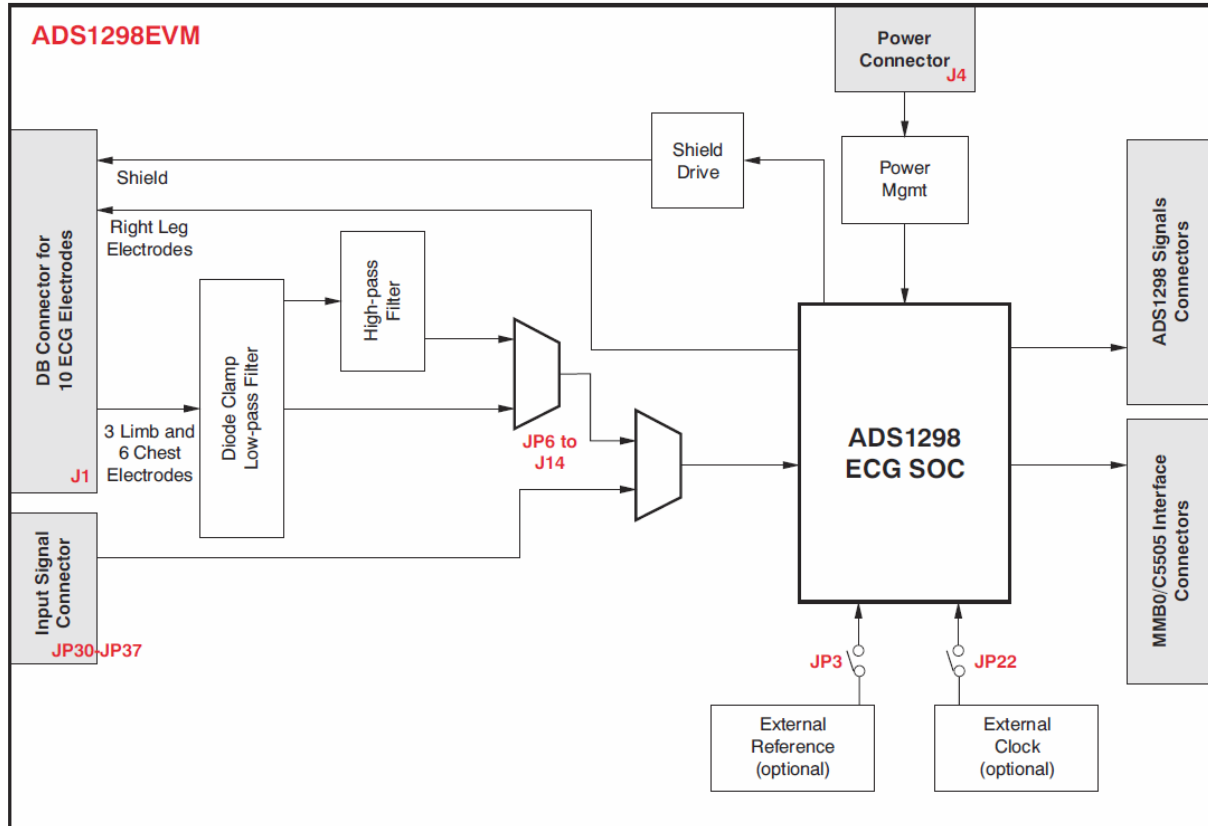


Figure 7. ADS1298ECG-FE Front-End Block Diagram

Table 6. Test Signals

Signal	J6 Pin Number		Signal
PACEOUT1	1	2	PACEOUT2
RESERVE	3	4	RESERVE
GPIO4	5	6	PWDNB
GPIO3	7	8	DAISY_IN
EXT_CLK	9	10	GND

Table 7. Serial Interface Pinout

Signal	J3 Pin Number		Signal
START/CS	1	2	CLKSEL
CLK	3	4	GND
NC	5	6	GPIO1
CS	7	8	RESETB
NC	9	10	GND
DIN	11	12	GPIO2
DOUT	13	14	NC/START
DRDYB	15	16	NC
NC	17	18	GND
NC	19	20	NC

#### 4.6.2 Arbitrary Input Signals

Arbitrary input signals can be fed to the ADS1298 by bypassing the DB15 connector and feeding the signal directly at jumpers JP30-JP37. Remove the set of 16 jumpers at JP30-JP37. The signal must be fed in differentially because all channel inputs are differential. **If single-ended signals are used, bias the negative input of the channels to a mid-supply voltage.** Again, care must be taken to ensure that the single-ended signal has an offset equal to the voltage supplied at the negative input of the channel.

#### 7.2 Lead Derivation

The EVM is configured to generate 12 leads of ECG signals from the 10 electrodes using the eight channels of the ADS1298. **Two of the limb leads and the six chest leads are computed purely in the analog domain** (Leads I, II, V1, V2, V3, V4, V5 and V6). The augmented leads and Lead III are computed digitally. The channel assignments are described in Table 26.

**Table 26. Lead Generations**

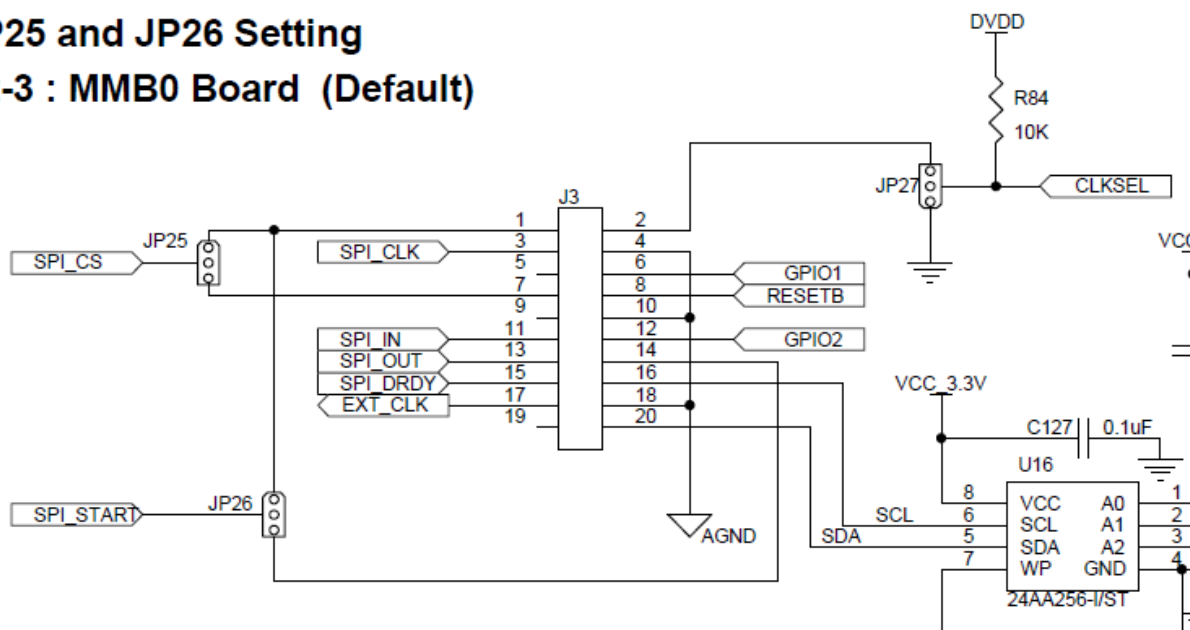
ADS1298 Input Channels	Derived Lead <sup>(1)</sup>
1	V6 = V6 – WCT
2	LI = LA – RA
3	LII = LL – RA
4	V2 = V2 – WCT
5	V3 = V3 – WCT
6	V4 = V4 – WCT
7	V5 = V5 – WCT
8	V1 = V1 – WCT

<sup>(1)</sup> WCT = (LA + RA + LL)/3

Summary of information to interface the MSP430 with the ADS1298. Info taken from TI Documentation and gathered from E2E forums hosted by TI. Author: Jesus Efrain Gaxiola-Sosa, efraingaxiola@gmail.com. **Disclaimer: The rights for the figures and text definitions showed on this document belong to Texas Instruments® and are used for education purposes only.**

## JP25 and JP26 Setting

### 2-3 : MMB0 Board (Default)



This is connected to J4 of the MMB0. Some pins are directly connected and others through the SN74CB3T3245, a TTL to CMOS converter.

Pin	Name	J3 on ADS1298 Board	J4 on MMB0-TMS320V
1	SPI_CS/START	SPI Chip Select, active low/Start Conversion	CNTL--GPIO6
2	CLKSEL_Jumper	Master CLK select	DCGPIO0---GPIO0
3	SPI_CLK	SPI Clock	CLKX-CLKX1
4	AGND	Analog GND	DGND
5	NC	Not connected	CLKR—CLKR1
6	GPIO1	General-purpose I/O 1	DCGPIO1---GPIO1
7	SPI_CS_Jumper	SPI Chip Select, active low	FSX—FSX1
8	RESETB	System RESET, active low	DCGPIO2---GPIO2
9	NC	Not connected	FSR-FSR1
10	AGND	Analog GND	DGND
11	SPI_IN	SPI data input	DX--DX1
12	GPIO2	General-purpose I/O 1	DCGPIO3---GPIO3
13	SPI_OUT	SPI data out	DR—DR1
14	SPI_START	Start conversion	DCGPIO4---GPIO4
15	SPI_DRDY	SPI data ready, active low	INT--INT4
16	SCL	EEPROM Serial CLK	DSCL—SCL/PACA95
17	EXT_CLK	External Master CLK input or internal master clock Output	DCTOUT-- TIN/TOU0
18	AGND	Analog GND	DGND
19	NC	Not connected	DCGPIO5---GPIO5
20	SDA	EEPROM Serial data	DSDA-SDA/PACA95

Highlighted pins are to be connected to MSP430

Summary of information to interface the MSP430 with the ADS1298. Info taken from TI Documentation and gathered from E2E forums hosted by TI. Author: Jesus Efrain Gaxiola-Sosa, efraingaxiola@gmail.com. **Disclaimer: The rights for the figures and text definitions showed on this document belong to Texas Instruments® and are used for education purposes only.**

Pin	Name	J3 on ADS1298 Board	ON MSP430 (Jumper in board)
1	SPI_CS/START	SPI Chip Select, active low/Start Conversion	SPI_CS/P4.2(J14)
2	CLKSEL_Jumper	Master CLK select	GPIOX/P1.2GDO0(J10)
3	SPI_CLK	SPI Clock	SPI_CLK/P4.5(J16)
4	AGND	Analog GND	DGND
6	GPIO1	General-purpose I/O 1	GPIOX/P1.3-GDO2 (J12)
8	RESETB	System RESET, active low	GPIOX/P1.4-(J5)
11	SPI_IN	SPI data input	SPI_IN/P4.4(SOMI-J20)
12	GPIO2	General-purpose I/O 1	GPIOXP/1.5-(J3)
13	SPI_OUT	SPI data out	SPI_OUT/P4.3(SIMO-J18)
14	SPI_START	Start conversion	GPIOX/P1.6-(J6)
15	SPI_DRDY	SPI data ready, active low	GPIOX/P1.7-(J8)

## Summary from forums

With the MSP430 you would set it up in 3-wire mode and use a GPIO function to control the chip select. to the [ADS1298](#). You can enter a 'write' loop which would send three (or more) 8-bit bytes via the SPI port to the [ADS1298](#). Toggle /CS low, write the desired data, then toggle /CS high... The registers are configured with the WREG command as described in Table 11 on page 37 of the data sheet.

[http://e2e.ti.com/support/data\\_converters/precision\\_data\\_converters/f/73/t/56258.aspx](http://e2e.ti.com/support/data_converters/precision_data_converters/f/73/t/56258.aspx)

## PCB design and noise

[http://e2e.ti.com/support/data\\_converters/precision\\_data\\_converters/f/73/p/203978/726138.aspx#726138](http://e2e.ti.com/support/data_converters/precision_data_converters/f/73/p/203978/726138.aspx#726138)

## MSP430 ADS1298 and USB interface

[http://e2e.ti.com/support/data\\_converters/precision\\_data\\_converters/f/73/t/151738.aspx](http://e2e.ti.com/support/data_converters/precision_data_converters/f/73/t/151738.aspx)

## MSP430FG4618 and ADS1298

[http://e2e.ti.com/support/data\\_converters/precision\\_data\\_converters/f/73/t/72825.aspx](http://e2e.ti.com/support/data_converters/precision_data_converters/f/73/t/72825.aspx)

## ADS1298 Code

[http://e2e.ti.com/support/data\\_converters/precision\\_data\\_converters/f/73/p/60104/214025.aspx#214025](http://e2e.ti.com/support/data_converters/precision_data_converters/f/73/p/60104/214025.aspx#214025)

Summary of information to interface the MSP430 with the ADS1298. Info taken from TI Documentation and gathered from E2E forums hosted by TI. Author: Jesus Efrain Gaxiola-Sosa, efraingaxiola@gmail.com. **Disclaimer: The rights for the figures and text definitions showed on this document belong to Texas Instruments® and are used for education purposes only.**

Code questions

[http://e2e.ti.com/support/data\\_converters/precision\\_data\\_converters/f/73/t/97138.aspx#339214](http://e2e.ti.com/support/data_converters/precision_data_converters/f/73/t/97138.aspx#339214)

Source code from client is evaluated

[http://e2e.ti.com/support/data\\_converters/precision\\_data\\_converters/f/73/t/50154.aspx](http://e2e.ti.com/support/data_converters/precision_data_converters/f/73/t/50154.aspx)

SPI Code debugging

[http://e2e.ti.com/support/data\\_converters/precision\\_data\\_converters/f/73/t/87971.aspx](http://e2e.ti.com/support/data_converters/precision_data_converters/f/73/t/87971.aspx)

ADS1298EVM noise problem

[http://e2e.ti.com/support/data\\_converters/precision\\_data\\_converters/f/73/p/60434/217707.aspx#217707](http://e2e.ti.com/support/data_converters/precision_data_converters/f/73/p/60434/217707.aspx#217707)

ADS1298 with DSP, SPI issue

[http://e2e.ti.com/support/data\\_converters/precision\\_data\\_converters/f/73/p/230629/809964.aspx#809964](http://e2e.ti.com/support/data_converters/precision_data_converters/f/73/p/230629/809964.aspx#809964)

Software installation

[http://e2e.ti.com/support/data\\_converters/precision\\_data\\_converters/f/73/t/164743.aspx](http://e2e.ti.com/support/data_converters/precision_data_converters/f/73/t/164743.aspx)

[http://e2e.ti.com/support/data\\_converters/precision\\_data\\_converters/f/73/p/247226/877167.aspx](http://e2e.ti.com/support/data_converters/precision_data_converters/f/73/p/247226/877167.aspx)

[http://e2e.ti.com/support/data\\_converters/etc\\_data\\_converters/f/77/t/254639.aspx](http://e2e.ti.com/support/data_converters/etc_data_converters/f/77/t/254639.aspx)

[http://e2e.ti.com/support/data\\_converters/precision\\_data\\_converters/f/73/t/62393.aspx](http://e2e.ti.com/support/data_converters/precision_data_converters/f/73/t/62393.aspx)

[http://e2e.ti.com/support/data\\_converters/precision\\_data\\_converters/f/73/t/228515.aspx?pi73718=2](http://e2e.ti.com/support/data_converters/precision_data_converters/f/73/t/228515.aspx?pi73718=2)

[http://e2e.ti.com/support/data\\_converters/precision\\_data\\_converters/f/73/t/228515.aspx?pi73718=2](http://e2e.ti.com/support/data_converters/precision_data_converters/f/73/t/228515.aspx?pi73718=2)

[http://e2e.ti.com/support/data\\_converters/etc\\_data\\_converters/f/77/p/236315/828237.aspx#828237](http://e2e.ti.com/support/data_converters/etc_data_converters/f/77/p/236315/828237.aspx#828237)

<http://e2e.ti.com/search/default.aspx?q=ADS1298ECGFE-PDK+windows+8&g=45>

[http://e2e.ti.com/support/data\\_converters/precision\\_data\\_converters/f/73/p/69359/594546.aspx#594546](http://e2e.ti.com/support/data_converters/precision_data_converters/f/73/p/69359/594546.aspx#594546)

[http://e2e.ti.com/support/data\\_converters/precision\\_data\\_converters/f/73/t/248114.aspx](http://e2e.ti.com/support/data_converters/precision_data_converters/f/73/t/248114.aspx)

install any driver bypassing warning from WIN8

<http://libusb.6.n5.nabble.com/Windows-8-Testing-of-libusb-win32-and-libusbK-td5710793.html>

<http://sourceforge.net/projects/libwidi/files/zadig/>

Summary of information to interface the MSP430 with the ADS1298. Info taken from TI Documentation and gathered from E2E forums hosted by TI. Author: Jesus Efrain Gaxiola-Sosa, efraingaxiola@gmail.com. **Disclaimer: The rights for the figures and text definitions showed on this document belong to Texas Instruments® and are used for education purposes only.**

## **To bypass windows security and install ADS1298 usb drivers on windows 8 do the next:**

1. Windows Key + R
2. Enter shutdown.exe /r /o /f /t 00
3. Click the "OK" button
4. System will restart to a "Choose an option" screen
5. Select "Troubleshoot" from "Choose an option" screen
6. Select "Advanced options" from "Troubleshoot" screen
7. Select "Windows Startup Settings" from "Advanced options" screen
8. Click "Restart" button
9. System will restart to "Advanced Boot Options" screen
10. Select "Disable Driver Signature Enforcement"
11. Once the system starts, install the Arduino drivers as you would on Windows 7

Steps to install driver after Driver Signature Enforcement has been disabled:

1. Right click the Unknown Device in Device Manager
2. Select Update Driver Software
3. Select Browse my computer for software
4. Click Browse button
5. Select the Drivers folder under the location of your Arduino software installation. (Do not select the FTDI folder inside the Drivers folder)
6. Click OK
7. Click Next

### **ADS1298 FAQs**

[http://e2e.ti.com/support/data\\_converters/precision\\_data\\_converters/w/design\\_notes/1296.frequently-asked-ads1298-and-ads1198-questions.aspx](http://e2e.ti.com/support/data_converters/precision_data_converters/w/design_notes/1296.frequently-asked-ads1298-and-ads1198-questions.aspx)

more code from clients

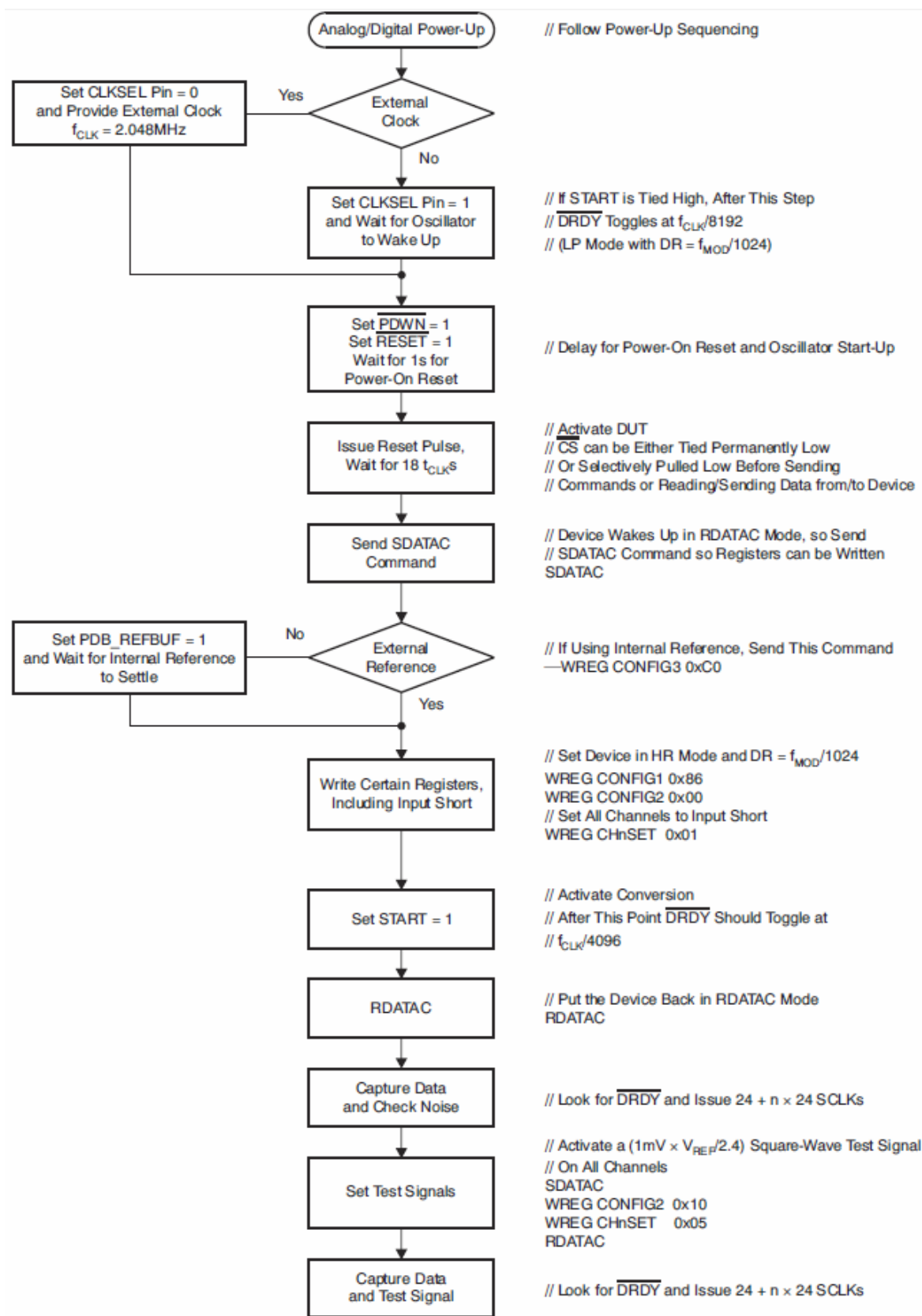
[http://e2e.ti.com/support/data\\_converters/precision\\_data\\_converters/f/73/p/122826/438630.aspx#438630](http://e2e.ti.com/support/data_converters/precision_data_converters/f/73/p/122826/438630.aspx#438630)

### **How to use motherboard MMB0 which ADS1298 plug into?**

[http://e2e.ti.com/support/data\\_converters/precision\\_data\\_converters/f/73/t/200451.aspx](http://e2e.ti.com/support/data_converters/precision_data_converters/f/73/t/200451.aspx)



Summary of information to interface the MSP430 with the ADS1298. Info taken from TI Documentation and gathered from E2E forums hosted by TI. Author: Jesus Efrain Gaxiola-Sosa, efraingaxiola@gmail.com. **Disclaimer: The rights for the figures and text definitions showed on this document belong to Texas Instruments® and are used for education purposes only.**



**Figure 77. Initial Flow at Power-Up**

Summary of information to interface the MSP430 with the ADS1298. Info taken from TI Documentation and gathered from E2E forums hosted by TI. Author: Jesus Efrain Gaxiola-Sosa, efraingaxiola@gmail.com. **Disclaimer: The rights for the figures and text definitions showed on this document belong to Texas Instruments® and are used for education purposes only.**

**Source code provided by TI:**

[http://e2e.ti.com/support/data\\_converters/precision\\_data\\_converters/f/73/t/60104.aspx](http://e2e.ti.com/support/data_converters/precision_data_converters/f/73/t/60104.aspx)

**init\_ADS1x9x.c, ADS1x9x\_Device\_Status.c, ADS1x9x\_Functions.c, and ADS1x9x\_SPI\_Functions.c:** These files provide functions for initializing and operating the [ADS1298](#). These files should give you a good idea of the flow required to configure the device.

**F5438\_ADS1x9x\_Device\_Interface.c, F5438\_Modular\_EVM\_IO\_Functions.c:** These files provide some of the application-specific functions that support the code in the ADS1x9x files. This is just an example of how we implemented it on our [MSP430F5438](#) board, ***you will need to setup the pins, interrupts, etc. for your specific application.***

**ADS1x9x\_Main\_Shell.c:** This is a shell for the main file of the project. It should give you an idea of how the functions provided in the above files are used to operate the device.

[3225.ADS1298\\_Code.zip](#)

**init\_ADS1x9x.c**

Set's up an ADS1x9x device. All initialization calls are initiated from main.c

**ADS1x9x\_Device\_Status.c**

Status handling for ADS1x9x devices. All initialization calls are initiated from main.c

**ADS1x9x\_Functions.c**

Sets up the Functions for Operating the ADS1x9x. All initialization calls are initiated from main.c

**ADS1x9x\_SPI\_Functions.c**

SPI Communication for ADS Devices. All initialization calls are initiated from main.c

**F5438\_ADS1x9x\_Device\_Interface.c**

Interface for ADS1x9x devices. All initialization calls are initiated from main.c

**F5438\_Modular\_EVM\_IO\_Functions.c**

Set's up and controls the I/O (GPIOs and LEDs) for the Modular EVM Board

**ADS1x9x\_Main\_Shell.c**

Set's up the MSP430 uC Peripherals. All calls are initiated from this routine. While Loop cycles 65,535 times before putting the MCU in Sleep mode. ISR must pull Sleep mode out of the Status Register to restart Idle\_count