TI Designs

MultiParameter BioSignal Monitor Design Guide

TI Designs

TI Designs provide the foundation that you need including methodology, testing and design files to quickly evaluate and customize the system. TI Designs help *you* accelerate your time to market.

Design Resources

TIDM-BIOSIGNMONITOR	Tool Folder Containing Design Files
MSP430FR5989	Product Folder
LMP2231	Product Folder
TPS730	Product Folder
ADS1292R	Product Folder
TPS61220	Product Folder
TMP112	Product Folder
RF430CL330h	Product Folder
LM4041-N	Product Folder

TI E2E™ Community

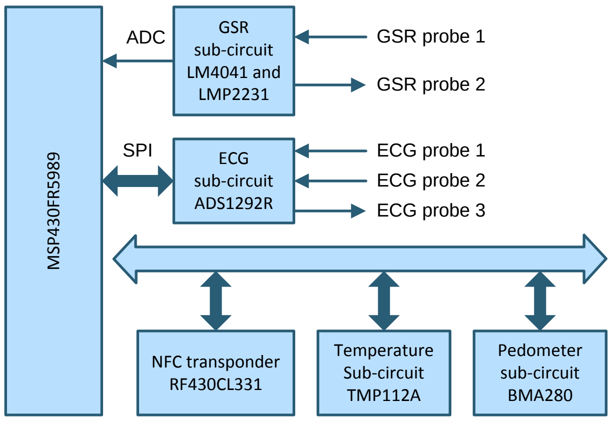
ASK Our E2E Experts
WEBENCH® Calculator Tools

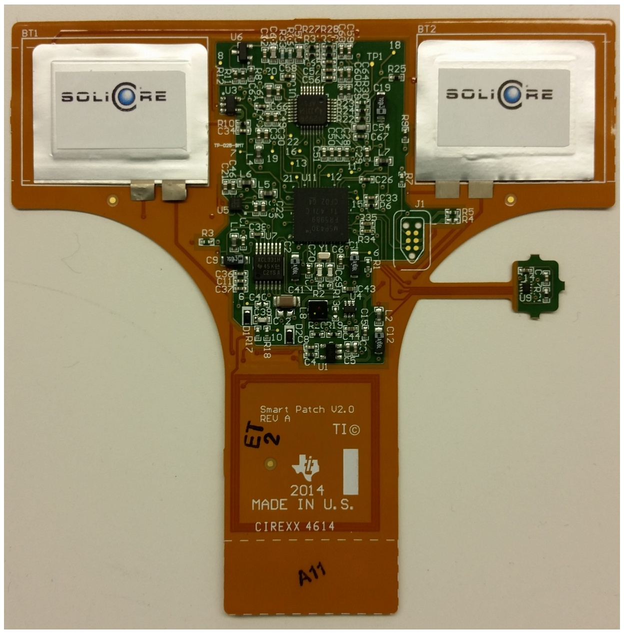
Design Features

- Highly Integrated Multiparameter Biosignal Sensors
- Capable Of Acquiring ECG, GSR, Respiration, Motion And Temperature Signals
- NFC-Enabled Real-Time Bi-Directional Communication
- Ultra-Low-Power Design
- Battery Powered For Continuous Data Logging
- Includes FW, Android™ App, HW design files, User's Guide, And Demo Quick Start Guide

Featured Applications

- Medical Devices
- Healthcare
- Fitness
- Activity Monitors
- Wearables
- Smart Clothing


MSP430FR5989
ADC
SPI
GSR sub-circuit LM4041 and LMP2231
GSR probe 1
GSR probe 2
ECG sub-circuit ADS1292R
ECG probe 1
ECG probe 2
ECG probe 3
NFC transponder RF430CL331
Temperature Sub-circuit TMP112A
Pedometer sub-circuit BMA280



An IMPORTANT NOTICE at the end of this TI reference design addresses authorized use, intellectual property matters and other important disclaimers and information.

General Texas Instruments High Voltage Evaluation (TI HV EVM) User Safety Guidelines

WARNING

Always follow TI's setup and application instructions, including use of all interface components within their recommended electrical rated voltage and power limits. Always use electrical safety precautions to help ensure your personal safety and those working around you. Contact TI's Product Information Center http://support/ti./com for further information.

Save all warnings and instructions for future reference.

Failure to follow warnings and instructions may result in personal injury, property damage, or death due to electrical shock and burn hazards.

The term TI HV EVM refers to an electronic device typically provided as an open framed, unenclosed printed circuit board assembly. It is **intended strictly for use in development laboratory environments, solely for qualified professional users having training, expertise and knowledge of electrical safety risks in development and application of high voltage electrical circuits. Any other use and/or application are strictly prohibited by Texas Instruments**. If you are not suitable qualified, you should immediately stop from further use of the HV EVM.

1. Work Area Safety
   (a) Keep work area clean and orderly.
   (b) Qualified observer(s) must be present anytime circuits are energized.
   (c) Effective barriers and signage must be present in the area where the TI HV EVM and its interface electronics are energized, indicating operation of accessible high voltages may be present, for the purpose of protecting inadvertent access.
   (d) All interface circuits, power supplies, evaluation modules, instruments, meters, scopes and other related apparatus used in a development environment exceeding 50Vrms/75VDC must be electrically located within a protected Emergency Power Off EPO protected power strip.
   (e) Use stable and nonconductive work surface.
   (f) Use adequately insulated clamps and wires to attach measurement probes and instruments. No freehand testing whenever possible.
2. Electrical Safety
   As a precautionary measure, it is always a good engineering practice to assume that the entire EVM may have fully accessible and active high voltages.
   (a) De-energize the TI HV EVM and all its inputs, outputs and electrical loads before performing any electrical or other diagnostic measurements. Revalidate that TI HV EVM power has been safely de-energized.
   (b) With the EVM confirmed de-energized, proceed with required electrical circuit configurations, wiring, measurement equipment connection, and other application needs, while still assuming the EVM circuit and measuring instruments are electrically live.
   (c) After EVM readiness is complete, energize the EVM as intended.

   WARNING: WHILE THE EVM IS ENERGIZED, NEVER TOUCH THE EVM OR ITS ELECTRICAL CIRCUITS AS THEY COULD BE AT HIGH VOLTAGES CAPABLE OF CAUSING ELECTRICAL SHOCK HAZARD.
3. Personal Safety
   (a) Wear personal protective equipment (for example, latex gloves or safety glasses with side shields) or protect EVM in an adequate lucent plastic box with interlocks to protect from accidental touch.

Limitation for safe use:

EVMs are not to be used as all or part of a production unit.

1 Key System Specifications

Table 1. Specifications

PARAMETER	SPECIFICATION	DETAILS
NFC Antenna	Inductance of the antenna	See Section 6.4

2 System Description

Biosignals are signals inherently produced by the human body and can be recorded by electronic equipment. Vital signs help determine the physiological status of the human body and help assess the general physical health of a person. The four primary vital signs are temperature, heart rate, respiration rate, and blood pressure. The Multiparameter Biosignal Monitor (MPBSM) is a proof-of-concept Evaluation Module (EVM) that is capable of measuring three of the four primary vital signs. The MPBSM proof of concept can directly measure Electrocardiogram (ECG), Galvanic Skin Response (GSR), and skin temperature. The MPBSM proof of concept can also indirectly measure heart rate, respiration rate, cadence, step count, and sweat-loss rate.

The MPBSM is classified as a wearable technology because it was designed using flexible substrates to mimic the appearance of a system bump: the system is placed on the chest diagonally across the heart using skin-safe adhesive tape. The MPBSM is a development platform that enables creating solutions to the problems associated with currently available wearable devices. For starters, the MPBSM provides a convenient all-in-one biosignal sensing system.

The MPBSM adheres to human skin, so it must be both durable and flexible. FR 7013 was used as the base substrate. Unlike FR-4, FR-1, CEM-1, or CEM-3, FR 7013 provides the durability and flexibility needed by the MPBSM. All of the electrical components on the MPBSM EVM are located on the top of the system with the exception of the probes needed by the GSR and ECG subcircuits, which are located on the bottom of the system. The probes are gold plated to reduce oxidation and increase conductance. Figure 1 illustrates the front of the system, and Figure 2 illustrates the back of the system.

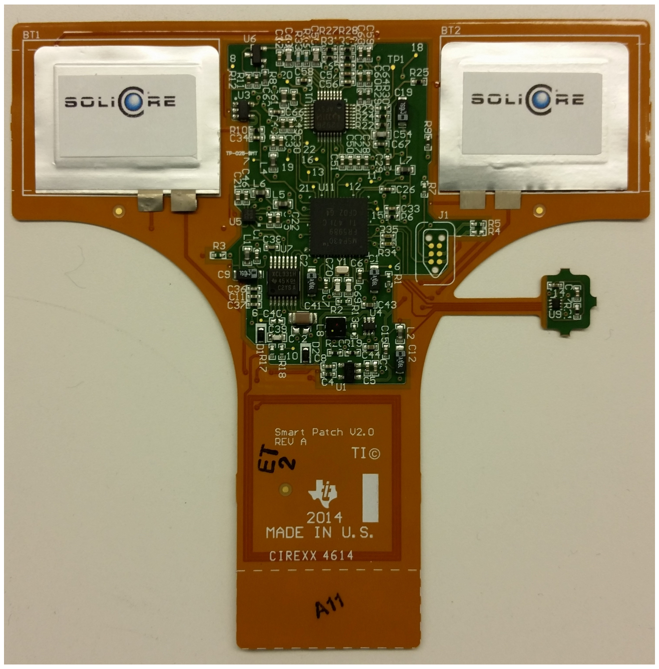


Figure 1. Multiparameter Biosignal Monitor EVM Front

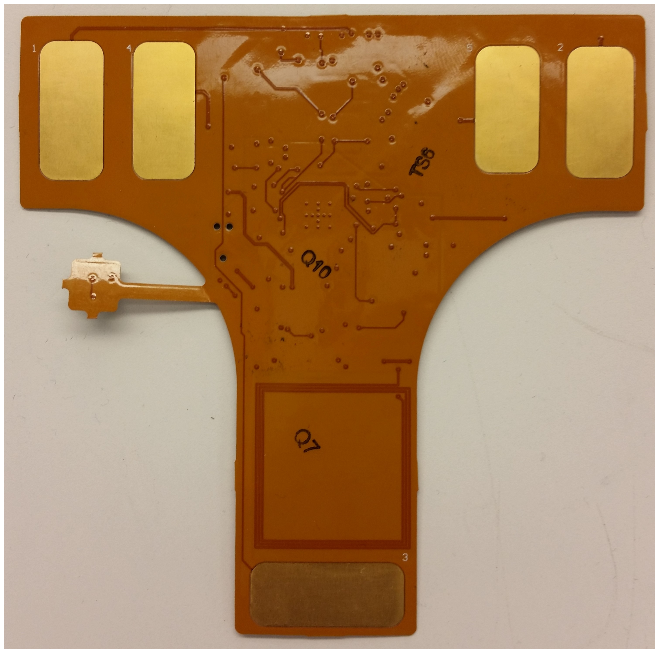


Figure 2. Multiparameter Biosignal Monitor EVM Back

To reduce power consumption, cost, and weight, and to increase flexibility and thereby comfort, the MPBSM does not incorporate a display. Due to the ubiquitous nature of today's cell phones, the MPBSM takes advantage and uses its screen as a display medium. The data gathered by the MPBSM sensors are transferred to the screen through NFC. NFC has been adopted by all major smart phone OEMs and is rapidly becoming a low-power alternative to Bluetooth® and Bluetooth low energy. The MPBSM app was developed for the Android operating system.

The relationship between the MPBSM IC sensors, software algorithms, and data displayed is illustrated in Figure 3. As can be seen from Figure 3, although the MPBSM is equipped with only four sensors, it can display nine health-data signs. Software algorithms help accomplish this process. For example, the ADS1292 measures ECG signals; however, in the process heart rate, respiration rate and leads off are also calculated.

Similarly, the GSR and temperature sensor data are combined to produce potential dehydration data. The BMA280 accelerometer calculates both cadence (steps per minute) and step count. As can be seen from Figure 3, the MPBSM system can display nine health data. The culmination of all the data displayed can be indicative of a person’s health state.

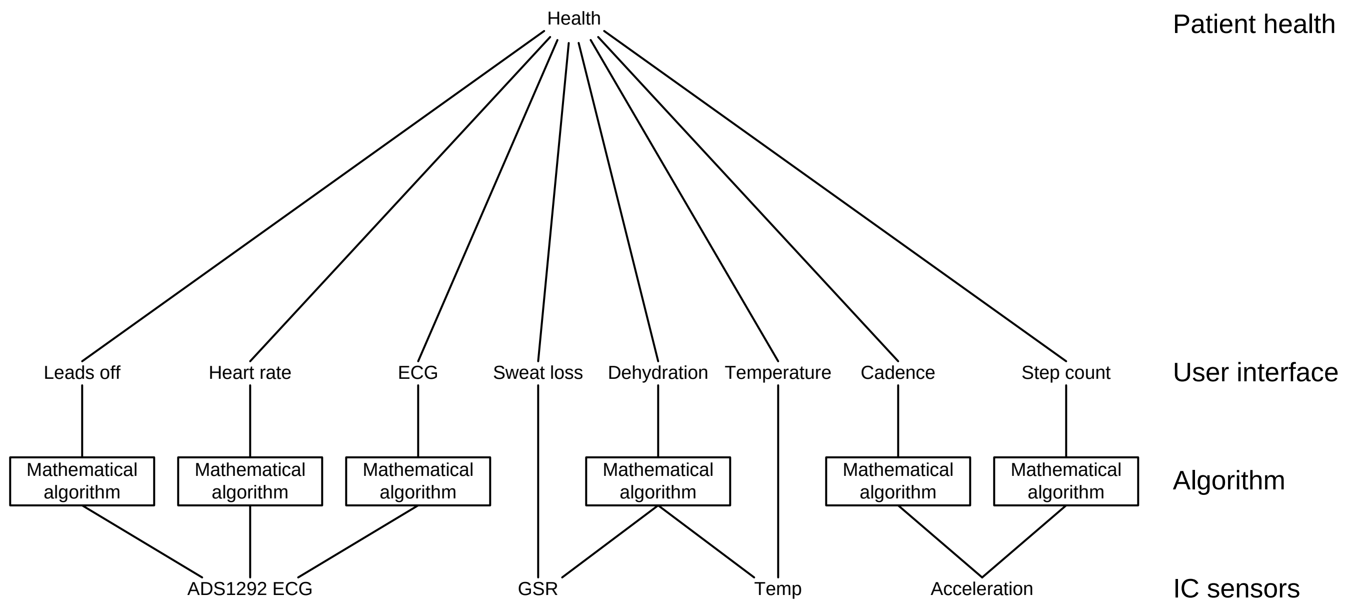


Figure 3. Multiparameter Biosignal Monitor Hardware / Software Structure

3 Block Diagrams

3.1 Highlighted Products

The MPSBM is primarily made up of eight sub-blocks: ECG, Temperature, Accelerometer, NFC transponder, GSR, MCU, JTAG, and power conversion. The size of these sub-blocks does not correspond to their complexity. Although not shown, each sub-block has a three-power supply source located on the bottom of Figure 4. Test points 4 and 5 located on the left side of Figure 4 represent the ECG leads. The test points 1, 2, and 3 on the right side correspond to the ECG leads. The following sections will go into great detail on each sub-block's subcircuit (see Figure 4).

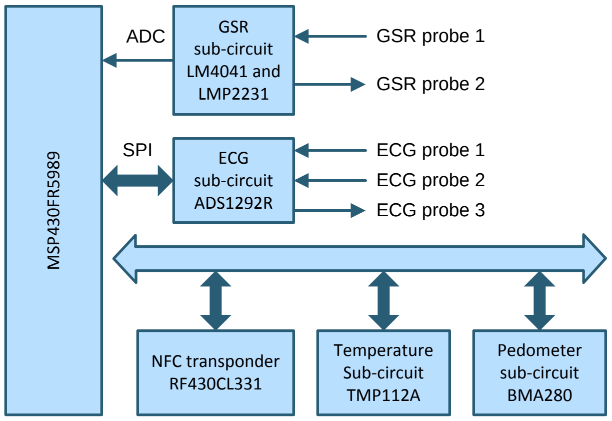

MSP430FR5989
ADC
GSR sub-circuit LM4041 and LMP2231
GSR probe 1
GSR probe 2
SPI
ECG sub-circuit ADS1292R
ECG probe 1
ECG probe 2
ECG probe 3
NFC transponder RF430CL331
Temperature Sub-circuit TMP112A
Pedometer sub-circuit BMA280

Figure 4. MPBSM Block Diagram

3.1.1 ADS1292R

This section provides a brief summary of the main elements of the ADS1292R analog front end.

The ECG subcircuit is implemented with the use of the ADS1292 IC. Learn more about the ADS1292R IC on the datasheet. The ADS1292R can be clocked either by an internal oscillator that generates a 512-kHz clock, or it can be clocked externally through the CLK pin (pin 17). Each clocking method has its advantages and disadvantages. Although the external clock provides high accuracy, it requires additional external components; however, the internal clock requires fewer components, but it suffers from temperature dependent performance.

As mentioned in the datasheet, internal clocking is ideally suited for low-power, battery-operated systems. The internal oscillator is trimmed for accuracy at room temperature. The accuracy varies over the specified temperature rang. The only permissible external clock frequencies are 512 kHz or 2.048 MHz. The higher frequency option has been provided to allow the SPI to run at a higher speed. The ADS1292R uses the SPI communication protocol to communicate with the MCU.

At the heart of the ADS1292R IC is the 24-bit, delta sigma (ΔΣ) analog-to-digital converter. The theory of operation of the ΔΣ is depicted in its simplified block diagram (see Figure 5). The main components are the ΔΣ Modulator and the digital / decimation filter, which is composed of a digital filter and a decimator. As Bonnie Baker explained it, “The design of delta-sigma (ΔΣ) analog-to-digital converters (ADCs) is approximately three-quarters digital and one-quarter analog.” The modulator is on the analog side and is responsible for digitizing the analog input signal; the digital / decimation filter is the digital side and is responsible for both removing the unwanted noise and slowing down the output-data rate. The modulator on the ΔΣ found on the ADS1292R is a second-order type. The input to the modulator is the time-varying analog voltage. The modulator converts the analog input signal to a high-speed, single-bit, modulated pulse wave. The modulated pulse wave outputted by the modulator is fed into the digital / decimation filter. The digital filter found on the ΔΣ is a third-order low-pass sinc filter.

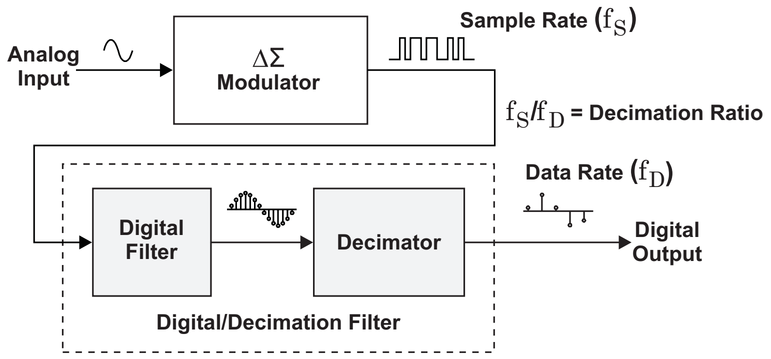


Figure 5. 24-Bit Delta-Sigma (ΔΣ) Analog-to-Digital Converter

The modulator is made out of two different amps, two integrators, a comparator (1-Bit ADC), and a feedback loop (see Figure 6). The modulator is basically a quantizer. The sample rate (fs) for the modulator found on the ADS1292 is fixed at 128 kHz. The only permissible external clock frequencies are 512 kHz or 2.048 MHz.

- If the external clock is 512 kHz, it must be scaled by a factor of 4.
- If the external clock is 2.048 MHz, it must be scaled by a factor of 16.

This scaling of the external clock is configured with the use of the LOFF_STAT register bit 6.

- If the external clock is 512 kHz, then LOFF_STAT register bit 6 is set to 0.
- If the external clock is 2.048 MHz, then LOFF_STAT register bit 6 is set to 1.

The internal oscillator generates a fixed 512-kHz clock.

One side effect of quantizers is quantization noise. One way to reduce unwanted noise caused by quantization is oversampling, which has the effect of decreasing the noise floor. Another effective technique for “reducing” unwanted noise is with the addition of an integrator, which causes noise shaping. The integrator acts as a low-pass filter to the input signal and a high-pass filter to the quantization noise. Thus, an integrator does not remove noise, but rather it shifts the quantized noise into the higher frequencies.

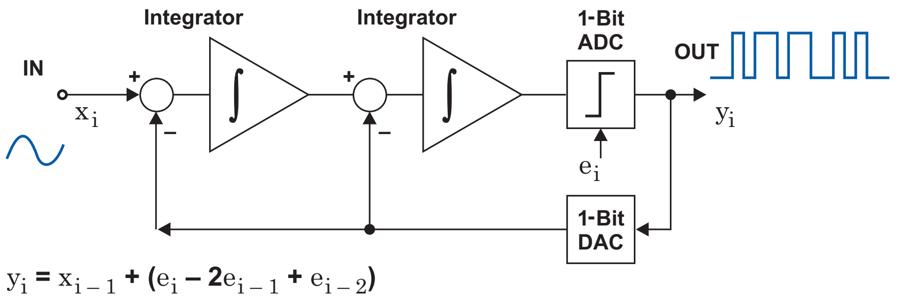


Figure 6. Simplified Modulator Block Diagram

Because the integrator found inside the modulator pushes floor noise to higher frequencies, and because the signal of interest is on the lower side of the frequency spectrum, a low-pass sinc filter is placed at the output of the integrator. The sinc filter is a variable-decimation-rate, third-order, low-pass filter that averages the 1-bit data stream and attenuates the high-frequency outside the band of interest noise. At this point, the output rate of a digital filter is the same as the sampling rate. Although the signal is a high-resolution digital version of the input signal, it is still far too fast to be useful. The decimator solves this problem by reducing the output-data rate to something more manageable; the decimator accomplishes this by discarding portions of the output data. The ADS1292 ΔΣ operates at data rates up to 8 kSPS.

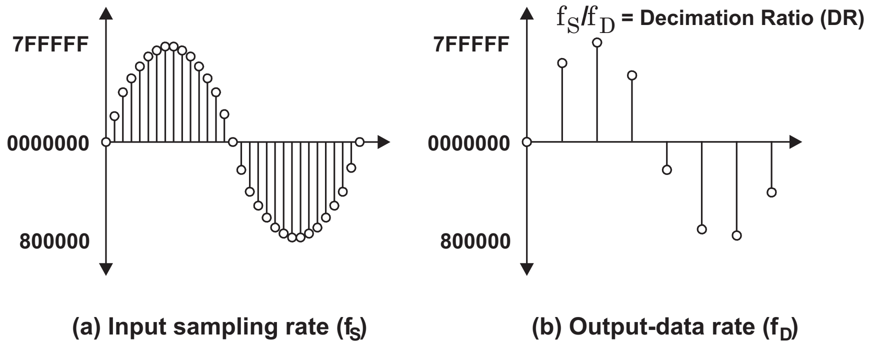

7FFFFF
0000000
800000
7FFFFF
0000000
800000
f_S/f_D = Decimation Ratio (DR)
(a) Input sampling rate (f_S)
(b) Output-data rate (f_D)

Figure 7. ΔΣ ADC Decimator Input / Output

For ECG monitoring to be effective, the recorded ECG signals must be clean and free of noise. Any distortion of the ECG signals caused by improper electrode-to-patient placement can lead to improper or missed diagnosis. Monitoring techniques should be implemented to verify that electrodes are properly adhered to the patient. The ADS1292R offers lead-off detection, which is a built-in monitoring circuitry that constantly monitors the ECG leads to ensure they are properly adhered to the patient's skin. At the application level, an alarm can sound when the ECG leads encounter any issues. With the ADS1292R, lead-off detection is implemented either by using an external pull-up or pull-down resistor, or the device internal current source or sink. An internal AC lead-off detection feature is also available. No external circuitry is needed to enable lead-off detection.

The ADS1292R version includes an integrated respiration circuitry that permits respiration rate calculation and detection (respiration rate is classified as one of the four vital signs). The ADS1292R is a two-channel device; however, if respiration rate detection is enabled, then Channel 1 can no longer measure ECG signals. Four pins are needed for respiration rate detection: IN1P, IN1N, RESP_MODP, and RESP_MODN.

The ECG subcircuit is located at the top of the system where the analog section is located (see Figure 8). The ECG subcircuit can measure only ECG signals; however, with mathematical manipulation, we can extract the respiration rate of an ECG signal. The ECG subcircuit requires three probes. The probes located at the top of the system are used to measure ECG signals and calculate respiration rate. The probe located at the bottom is used only for the calculation of ECG signals (see Figure 9).

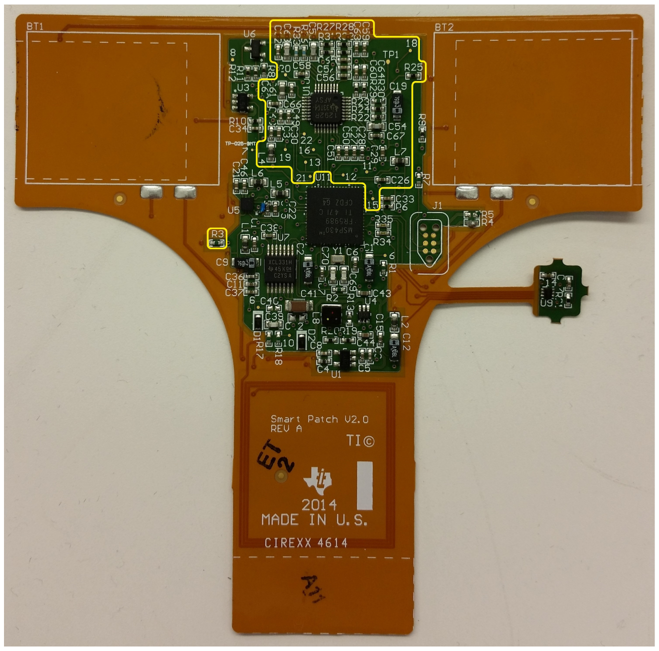


Figure 8. ECG Subcircuit—Hardware Location

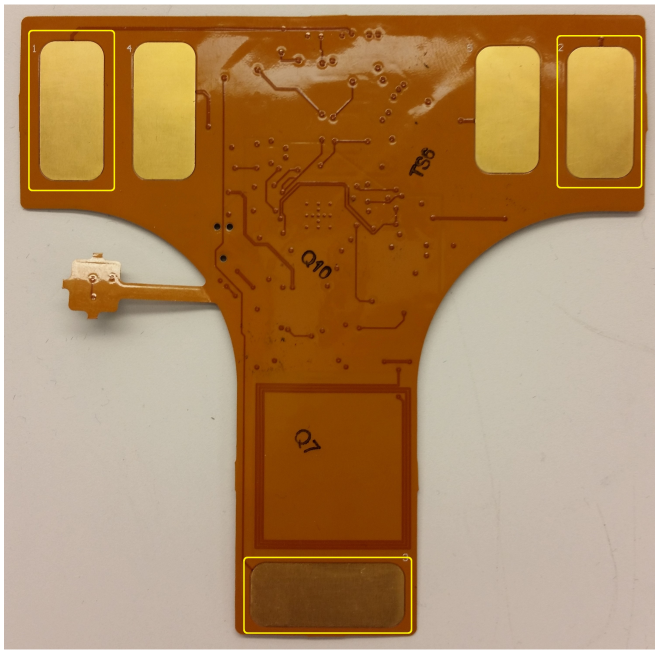


Figure 9. ECG Subcircuit—Probes System Back Location

The schematic for the ECG subcircuit shown in Figure 8 is illustrated in the following figures. The "nets" shown in Figure 10 are labeled ADS1292_ERA and ADS1292_ELA and correspond to the probes shown on the top of Figure 9, and the net labeled ADS1292_RLD corresponds to the bottom probe shown in Figure 9. The greater the number of channels an ECG monitoring system has, the better the ECG signal could be analyzed. ADS1292R is equipped with only two channels. As the schematic in Figure 10 shows in the MPBSM configuration, Channel 1 is used for respiration rate calculation, and Channel 2 is used for ECG measurement. The ECG signals produced by the body range in frequency between 0.01 – 300 Hz; as a result, a first-order low-pass filter is placed in between the probes and the ADS1292R IC.

Resistors R26 and R33 and capacitors C55 and C68 shown in Figure 10 form the single-state low-pass filter. ECG drift is also an issue faced by the ECG monitoring system and occurs when the ECG signals drift all over the monitoring device. Two techniques have been devised to reduce ECG drift: current injection and resistor biasing. For added security, the ECG subcircuit implements both methods to reduce ECG drift. The resistors R22, R23, R24, and R25 and capacitor C54 shown in Figure 11 form the RLD current injection method and resistors R27, R28, R31, and R32 shown in Figure 10 form the resistor biasing method. The ADS1292 decoupling capacitors are shown in Figure 12.

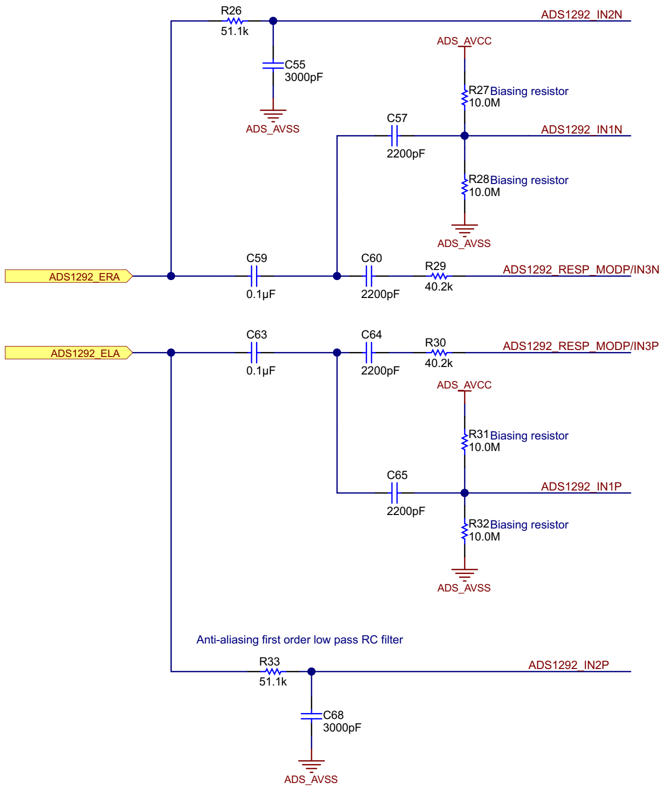


Figure 10. ECG Subcircuit—Leads

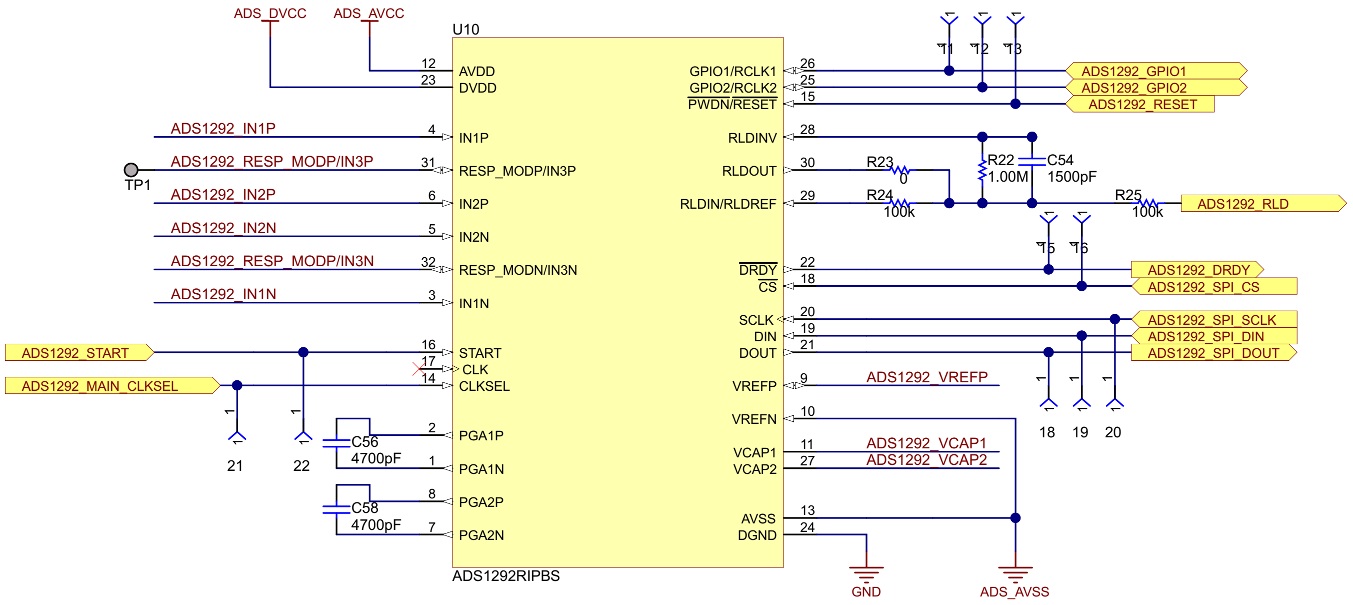


Figure 11. ECG Subcircuit—ADS1292 and RLD

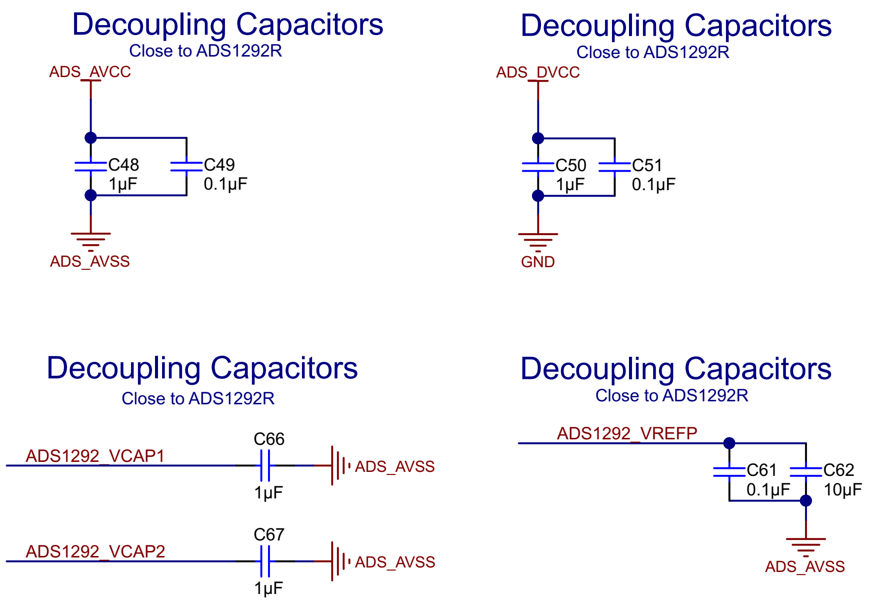


Figure 12. ECG Subcircuit—Decoupling Capacitors

3.1.2 LM4041& LMP2231

This section provides a brief summary of what can be found in the LMP2231 and LM4041 datasheets and should by no means supplement it.

The GSR subcircuit is implemented with the use of a LMP2231 single micro-power precision amplifier and an LM4041 precision micro-power shunt voltage reference. The LMP2231 and the LM4041 ICs are fully described on their datasheets (SNOS641 and LMP2231). The person's skin resistance is directly related to their perspiration level. Initially when a patient's skin is dry, their skin resistance is greater and lies in the Mega Ohms range. However, as a person begins to perspire, their skin resistance starts decreasing rapidly. The GSR subcircuit converts this resistance measurement into a voltage value. If the resistance is close to zero, then the output voltage is at its highest, which would correspond to 3 V. If the resistance is high, the output voltage is at its lowest, which would correspond to ~0 V. Therefore, the GSR subcircuit's output voltage can swing anywhere from 0 V to 3 V.

The output voltage swing of the GSR subcircuit lies within the acceptable MSP430™ ADC voltage input ratings, meaning that no additional signal conditioning hardware is needed in between the GSR subcircuit and the MSP430 ADC; its output is fed directly into the MCU ADC. Unlike the ECG subcircuit, which is partially digital and partially analog, the GSR is a completely analog circuit. To reduce parasitic noise inherent in all electrical components and to further increase the signal-to-noise ratio, all of the electrical components used to construct the GSR subcircuit are high precision with 0.5 percent tolerance or lower.

The location of the GSR sub-circuit within the MPBSM system is illustrated in Figure 13. As can be seen from Figure 13, the GSR subcircuit is located at the top of the system where the analog section is located. Unlike the ECG subcircuit, the GSR subcircuit requires only two probes. The location of the GSR probes is illustrated in Figure 14. As can be seen from Figure 14, the probes are located on the bottom of the system and furthermore they are the inner probes. Unlike ECG, the GSR subcircuit does not implement lead-off detection. The schematic for the GSR subcircuit shown in Figure 13 is illustrated in Figure 14. The "nets" shown in Figure 15 are labeled EL_GSR2 and EL_GSR1 and correspond to the probes shown on the top of Figure 14. As can be seen from Figure 15, the GSR subcircuit is a simple circuit made up of seven resistors, a capacitor, a shunt voltage reference, a high precision operations amplifier, and two electrodes that are represented by EL_GSR1 and EL_GSR 2. The ADS_AVCC provides the source voltage to the GSR subcircuit and is set to 3 V. At its peak, the GSR circuit consumes 120 µamps. The resistor-R8 and shunt-voltage-reference-U6 combo form the high-precision operations amplifier reference voltage. The voltage at node 1 is 1.03 V. The reference voltage determines how much current gets injected into the patient's skin when measuring the GSR. Anything below 500 millivolts is considered in the safe level.

Resistor R10 and capacitor C34 form the negative-feedback loop. The negative feedback is needed to provide a stable, accurate, and responsive output from the high-precision operations amplifier. The capacitor C34 is added to reduce the noise that can come from either the feedback circuit or from the EMI. Because both R10 and C34 are going to be in direct contact with a subjects skin, static electricity was an issue. To reduce the possibility of the MPBSM destruction caused by static electricity, R10 and C34 were chosen with high voltage values. Although this does not guarantee that a static electrical shock will not destroy the system, it should minimize the possibility of causing major harm.

As mentioned previously, the GSR is the electrical conductance across the skin, which varies depending on the amount of sweat present. Normal human skin resistance fluctuates anywhere from as high as 200 MΩ to as low as 50 Ω. Resistor R9 is a current limiting resistor; if the skin resistance decreases beyond a threshold value, then resistor R2 limits the current being injected into the subjects skin.

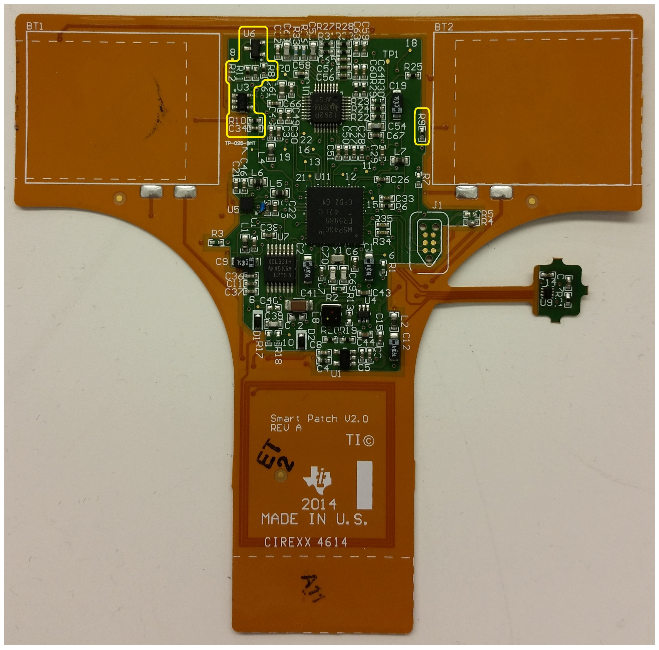


Figure 13. GSR Subcircuit—Hardware Location

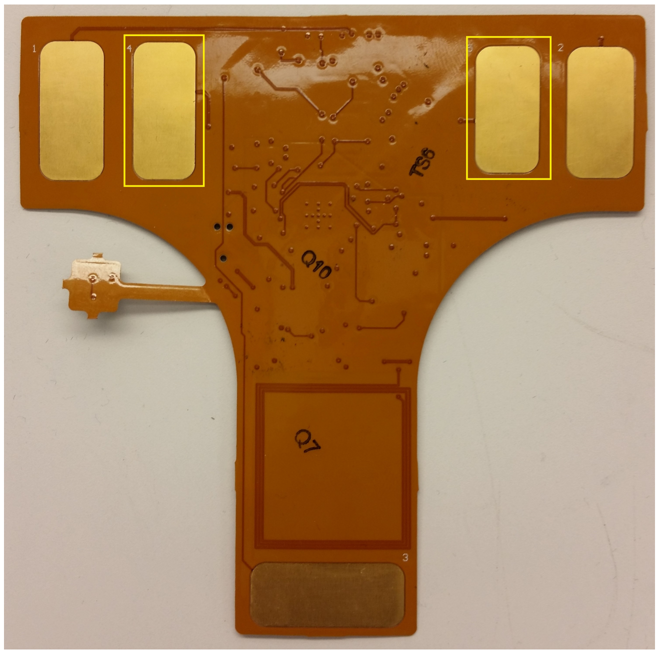


Figure 14. GSR Subcircuit—Probe Location

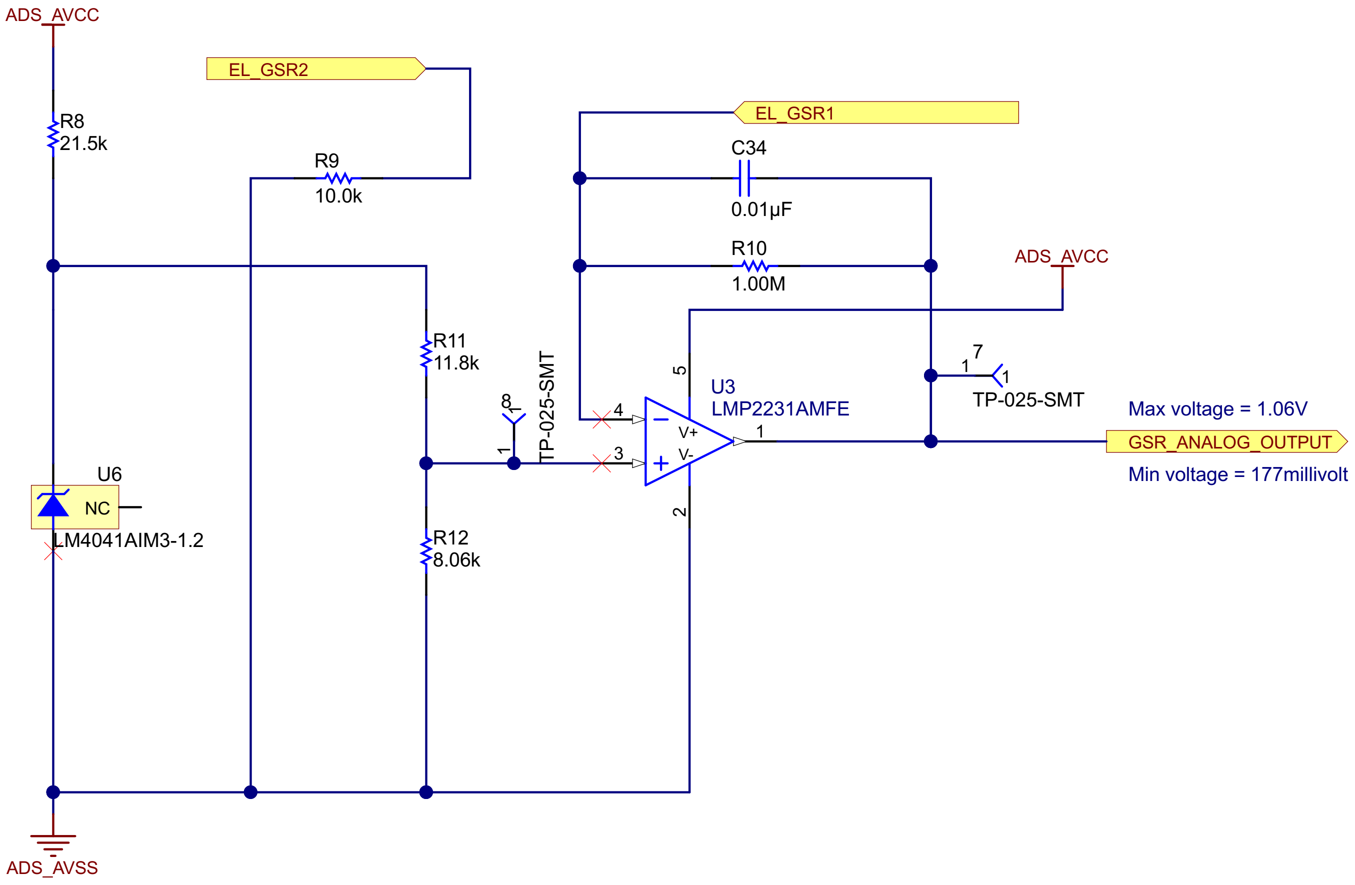


Figure 15. GSR Subcircuit—Schematic

3.1.3 BMA280

This section is only a very short summary of what can be found in the BMA280 datasheet and is not meant to supplement the datasheet.

The motion-sensing subcircuit is implemented with the use of the BMA280 3-axis accelerometer. The BMA280 is a triaxial, low-g acceleration sensor with digital output. Acceleration ranges for the BMA280 are ±2g, ±4g, ±8g, ±16g. The motion-sensing subcircuit is one of the simplest subcircuits found on the MPBSM and is only composed of seven components, four of which are for decoupling purposes.

The location of the motion-sensing subcircuit within the MPBSM system is illustrated in Figure 16. The motion-sensing subcircuit is considered a digital IC and is placed in the digital section of the system. As can be seen from Figure 16, the temperature subcircuit is located on the center-left side of the system. The placement of the BMA280 within the digital section was a matter of convenience. The motion-sensing subcircuit schematic is illustrated in Figure 17. Just like the TMP112A, the BMA280 has two configurable interrupt pins.

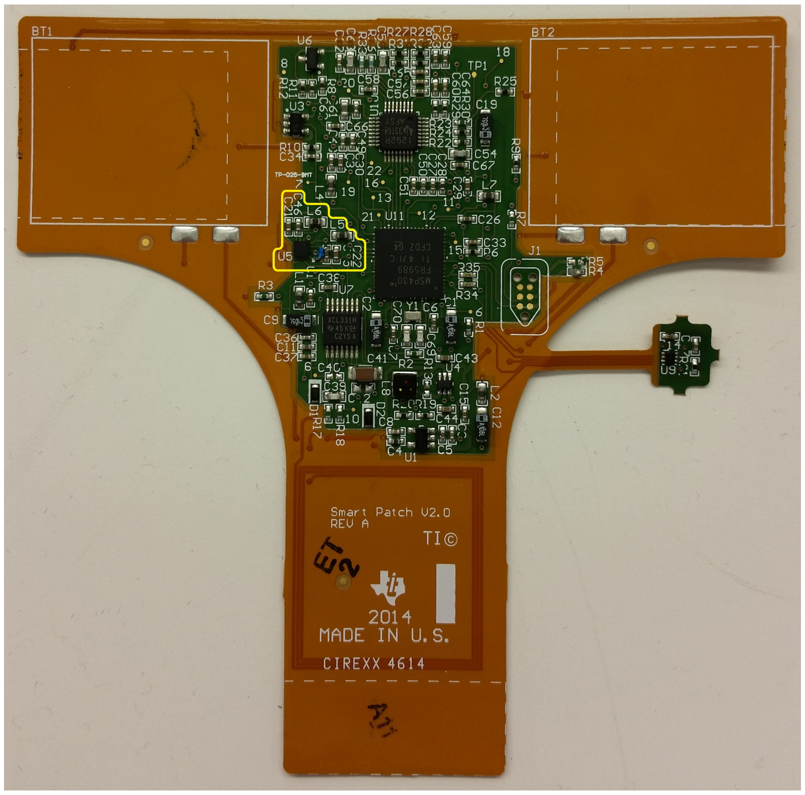


Figure 16. Motion Sensing Subcircuit—Location

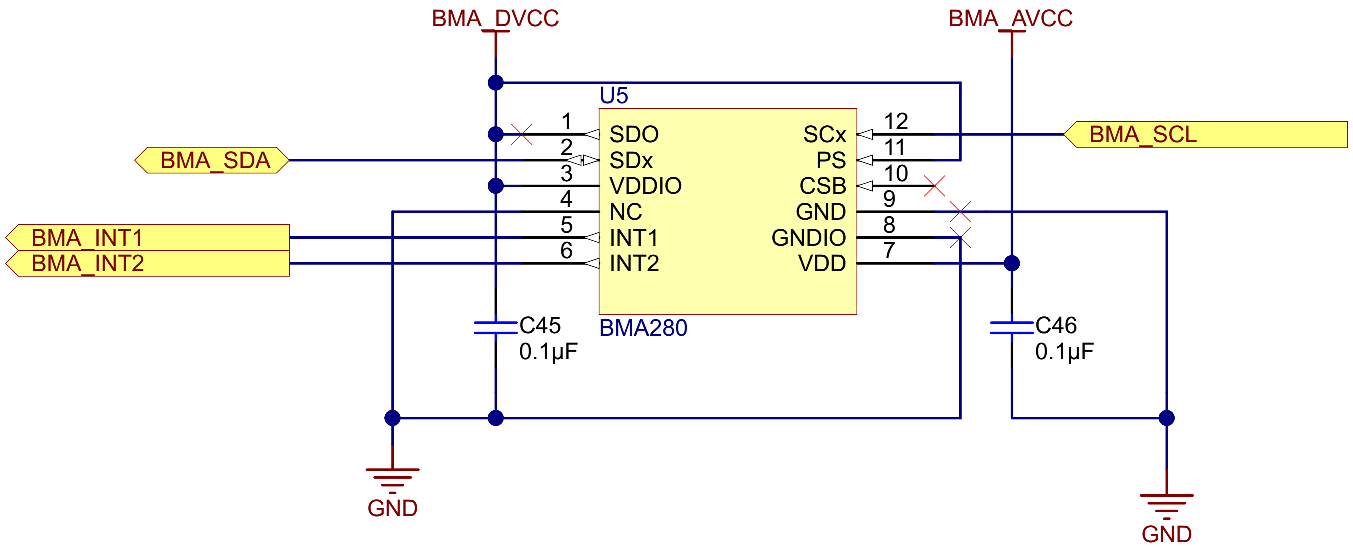


Figure 17. Motion Sensing Subcircuit—Schematic

3.1.4 TMP112A

This section provides a brief summary of the information contained in the TMP112A datasheet and is not meant to supplement the datasheet.

The temperature subcircuit is implemented with the use of the TMP112A, which is capable of reading temperatures to a resolution of 0.0625°C and can operate over a temperature range of –40°C to +125°C. The temperature subcircuit is one of the simplest subcircuits found on the MPBSM and is only composed of seven components, four of which are for decoupling purposes.

The location of the temperature subcircuit within the MPBSM system is illustrated in Figure 18. The temperature is considered a digital IC, and hence is placed in the digital section of the system. As can be seen from Figure 18 the temperature subcircuit is located on the right of the system and is composed of two subcircuits. The reason for this topology is to decrease thermal conduction by increasing temperature IC isolation. All IC devices generate heat; if the TMP112A is close enough those ICs, it will record the heat generated by those ICs instead of the skin temperature. To remedy this situation, the temperature IC protrudes from the main system to the point that heat conduction is not an issue. Before the patient places the MPBSM on his or her body, the extruded TMP112A IC sensor must be folded so that the IC is facing the patient's skin, which will maximize heat transfer between the patient's skin and TMP112A IC. The temperature subcircuit schematic and TMP112A configurable interrupt pins are shown in Figure 19.

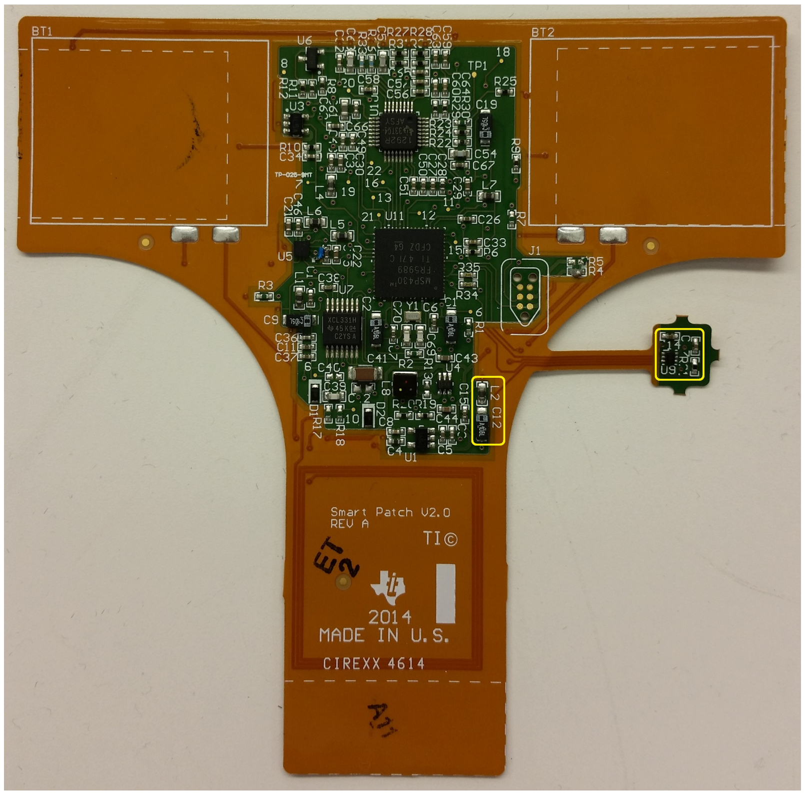


Figure 18. Temperature Subcircuit—Hardware Location

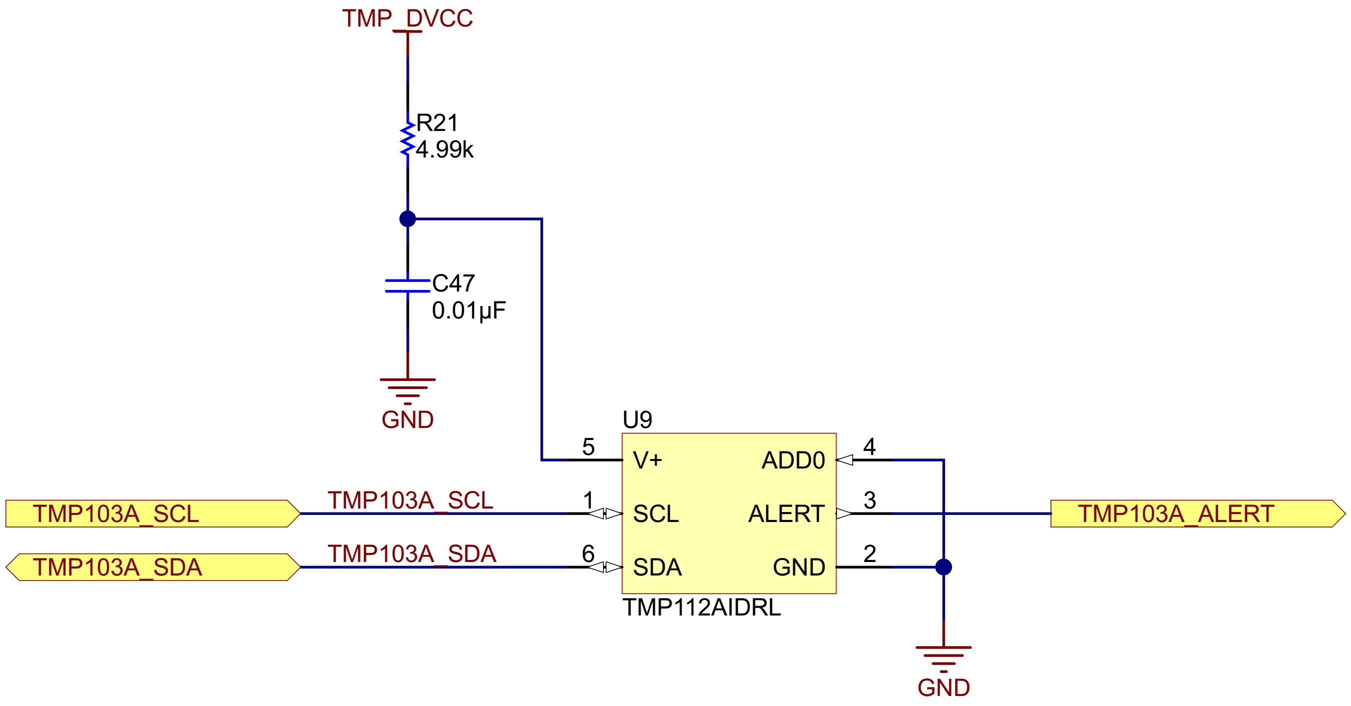


Figure 19. Temperature Subcircuit—Schematic

3.1.5 RF430CL331

This section provides a brief summary of the main elements of the RF430CL331. More information on the RF430CL331 IC can be found on the datasheet.

The NFC subcircuit is implemented with the use of the RF430CL331 IC. Like the ADS1292E, the RF430CL331 is partially analog and partially digital. For the purposes of the MPBSM system, the RF430CL331 is classified as a digital IC and is placed in the digital section. The location of the NFC transponder subcircuit within the MPBSM system is illustrated in Figure 20. As can be seen from Figure 20, the NFC subcircuit is purposefully placed towards the bottom of the MPBSM system.

A very important part of the NFC subcircuit is the antenna. Although the antenna is merely an inductor made out of a PCB trace, it allows the NFC subcircuit to communicate with a phone and, hence, the outside world. The antenna is not considered a traditional antenna in the sense that a piece of wire or a metal rod is sticking out. Instead, the antenna is a flat, rectangular PCB copper trace. The antenna forms part of a tank circuit that resonates at 13.56 MHz. The resonant frequency illustrated formula is illustrated in Equation 1. The capacitor is considered a tuning capacitor. Although the small-loop-inductance calculation program gives a good inductance estimate of the air coil, the measured inductance will be different. Because of this difference, it is a good idea to first manufacture one prototype, test it with the calculated L value, and see if it behaves to specifications. If the inductance is off by a few microhenries, the tuning capacitor can be used to compensate and reach the desired 13.56-MHz resonant frequency.

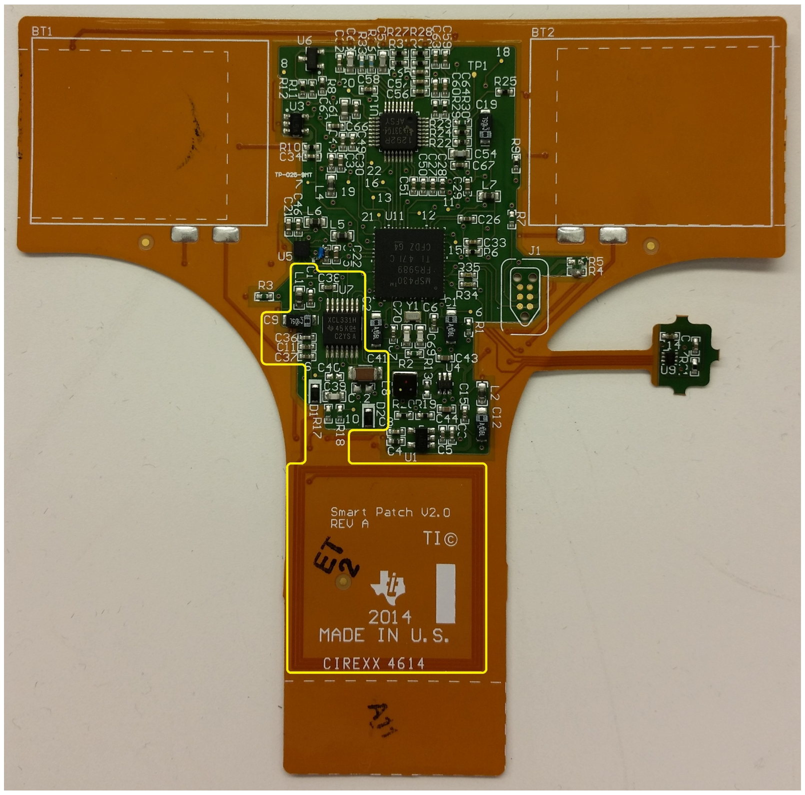
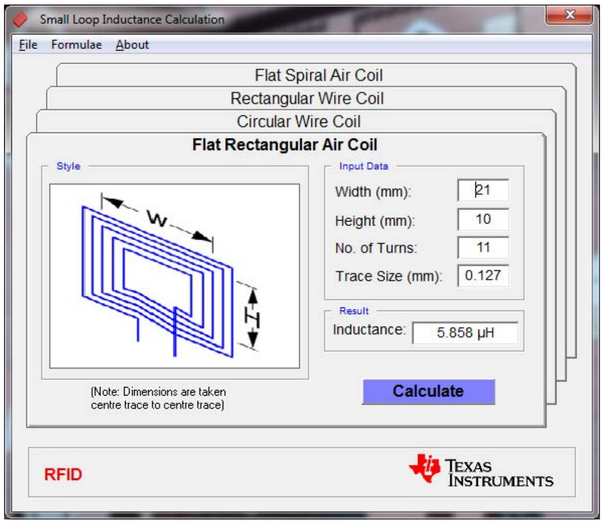


Figure 20. NFC Transponder Subcircuit—Schematic

$$f_0 = \frac{\omega_0}{2\pi} = \frac{1}{2\pi\sqrt{LC}} \tag{1}$$

The antenna works by way of induction, so it only works in the near field. The antenna inductance is directly determined by the PCB trace width, antenna perimeter, and number of PCB trace turns. By juggling those three parameters, virtually any reasonable-sized antenna can be created. The small-loop-inductance calculation program (shown in Figure 21) is used to calculate the inductance of the air coil inductor. The inductance depends on the trace width, height, number of turns, and trace size. Although the small-loop-inductance calculation is not available online, the formulas used by the program can be found here. Together the inductor and the capacitor form an LC circuit with its target resonant frequency set to 13.56 MHz. The resonant frequency illustrated formula is illustrated Equation 1.



Flat Rectangular Air Coil

Input Data

Width (mm): 21

Height (mm): 10

No. of Turns: 11

Trace Size (mm): 0.127

Result

Inductance: 5.858 µH

(Note: Dimensions are taken centre trace to centre trace)

Figure 21. Small-Loop-Inductance Calculation Program

Initially, the thought was to reduce the antenna size as much as possible; however, reading distance is affected. The bigger the antenna perimeter, the greater the tag-to-Android reading distances. One very obvious downside to large antennas is a potential increase in system size of the equipment. One very important design constraint regarding trace inductors is that there should be a 2-mm distance from electrical components including resister, capacitors, ground place, and others. Because of these factors, the antenna was strategically placed towards the bottom of the MPBSM. The NFC transponder subcircuit is illustrated in Figure 22.

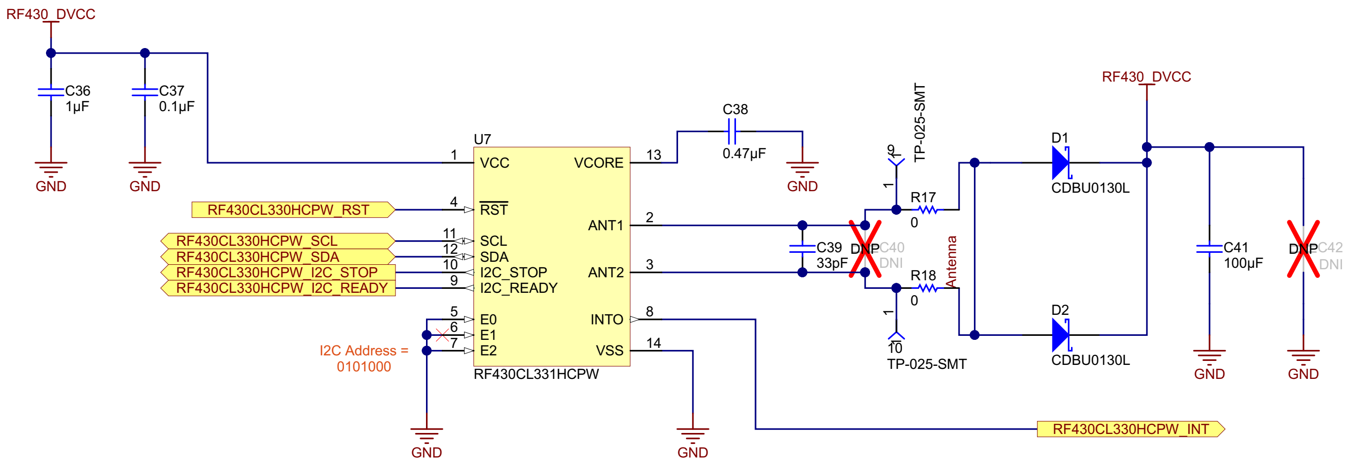


Figure 22. NFC Transponder—Schematic

4 System Design Theory

The Multiparameter Biosignal Monitor (MPBSM) uses nine Integrated Circuits (ICs). All of the ICs with exception of the BMA280 utilize Texas Instruments technology. Some of the ICs are for power conversion (TPS73030 and TPS6122), while others are sensors (ADS1292R, LMP2231, LM4041, BMA280, TMP112A) and provide the MPBSM system its sensing capabilities. The main ICs are the MCU (MSP430FR5989) and the NFC transponder (RF430CL331). The NFC transponder provides the MPBSM with the means to communicate with the outside world. The MCU is the central hub where all the sensor data is collected. Unlike traditional MCUs, the MSP430FR5989 is equipped with FRAM memory, which equates to single-chip data-logging capabilities. As mentioned previously, some of the collected sensor data are preprocessed by the MCU to produce virtual sensor data such as heart rate, cadence, and step count.

The simplified MPBSM block diagram is shown in Figure 23. As can be seen from Figure 23, the MPBSM sensor portion block diagram is composed mainly of four sensor subcircuits and a transceiver subcircuit, GSR subcircuit, temperature subcircuit, ECG subcircuit, accelerometer subcircuit, and NFC transponder antenna subcircuit. All commands given to the system and all responses sent back by the system are relayed through the NFC transponder. The transponder converts the NFC radio signals produced by the NFC phone into I^{2}C signals that the MCU can understand and vice versa. At the center of the simplified MPBSM sensor block diagram is the MSP430FR5989 MCU. Each subcircuit communicates with the MCU differently. The ADS1292E communicates with the MCU using SPI, while the BMA280, TMP112A, and the NFC transponder use I^{2}C. The GSR subcircuit that uses the LMP2231 and LM404 sends an analog signal to the on-board MCU ADC.

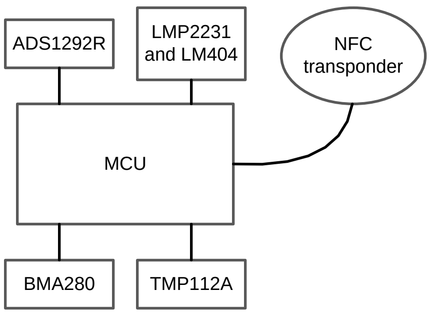


Figure 23. Multiparameter Biosignal Monitor EVM Simplified Block Diagram Sensor Section

Some of the MPBSM sensors are analog, while others are digital. It is a well-known fact that analog circuits are much more noise sensitive to digital circuits. Furthermore, it is also a well-known fact that digital circuits, because of all the fast switching that is occurring inside of them, generate a lot of electromagnetic interference (EMI) noise. The noise generated by digital circuits tends to be picked up by nearby analog circuits.

To keep noise disturbance to a minimum, several counter measures were taken during the design and development of the MPBSM system. First, the analog subcircuits were physically isolated from the digital subcircuits. In the MPBSM system, the analog sensors (GSR and ECG) are located on the top left, while the rest of the digital sensors (temperature, accelerometer, and NFC transponder) are located towards the bottom. In addition to physical isolation, ground isolation was also executed. The separation of analog ground from digital ground is illustrated in Figure 24. Most of the probes are located on the analog ground side, which makes sense because the probes measure analog signals, which are inherently produced by the body.

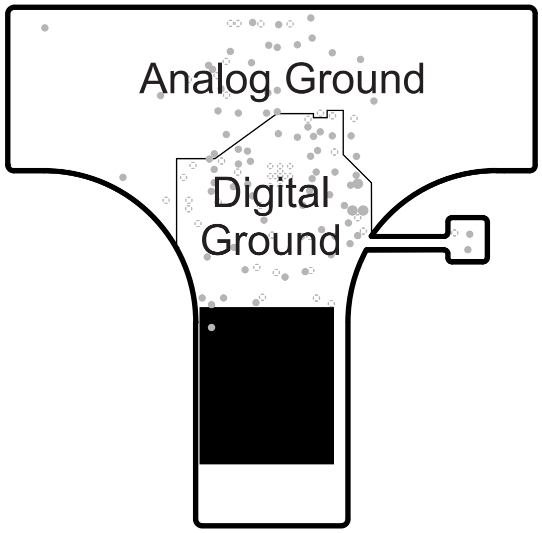


Figure 24. Multiparameter Biosignal Monitor EVM Block Analog and Digital Ground Planes

5 Getting Started Hardware

The MPBSM requires preparation before ECG reading can be conducted.

1. The MPBSM cannot be placed anywhere on the body
2. Conductive gel application is needed to reduce noise
3. The Android app must be installed.

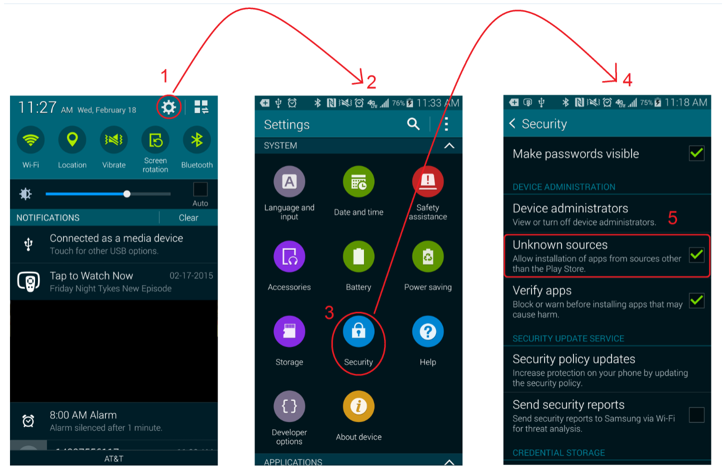
5.1 Firmware Download

To download the firmware to the MPBSM, a Tag-Connect Spy-Bi-Wire No-Legs (TC2030-MCP-NL(6”)) connector in conjunction with a MSP-FET Flash emulation too are needed. The reason why this topology programming method was used is because it saves overall PCB space and allows the MPBSM to be physically smaller.

5.2 Applications Installation

As of the writing of this document, the Smart Patch app is not available in the Android Play Store, so the app can only be installed manually. Follow these steps to manually install the Smart Patch app:

1. Configure the phone to allow installation of “Unknown Sources”. The procedure to reach the “Unknown Sources “setting is illustrated in Figure 25.



Make passwords visible ✓

DEVICE ADMINISTRATION

Device administrators
View or turn off device administrators.

Unknown sources ✓
Allow installation of apps from sources other than the Play Store.

Verify apps ✓
Block or warn before installing apps that may cause harm.

SECURITY UPDATE SERVICE

Security policy updates
Increase protection on your phone by updating the security policy.

Send security reports
Send security reports to Samsung via Wi-Fi for threat analysis.

CREDENTIAL STORAGE

Figure 25. Unknown Source Install Procedure

NOTE: The MPBSM app is only supported in the Android operating system.

The MPBSM app was developed on a Galaxy S5 and hence only tested on Galaxy S5. It is anticipated, however, that no issues should be encountered with other devices.

The MPBSM app has been designed and tested on Android version 4.4.2. Running the app on a handset running Android version 4.4.2 and below is not guaranteed to work.

---

2. Connect the Android mobile device to a computer using a USB A To USB B Micro Cable.
3. Copy the Android .apk to your Android device file system.
4. Keep track of the path of where the .apk is.
5. Navigate to the directory where the .apk was saved.
6. Once you navigate to the MPBSM.apk directory, launch the file. This should launch the install pop-up menu.
7. Click install and wait for the app to install. If successfully installed, an “App Installed” pop-up menu will appear. The manual app install procedure is illustrated in Figure 26.

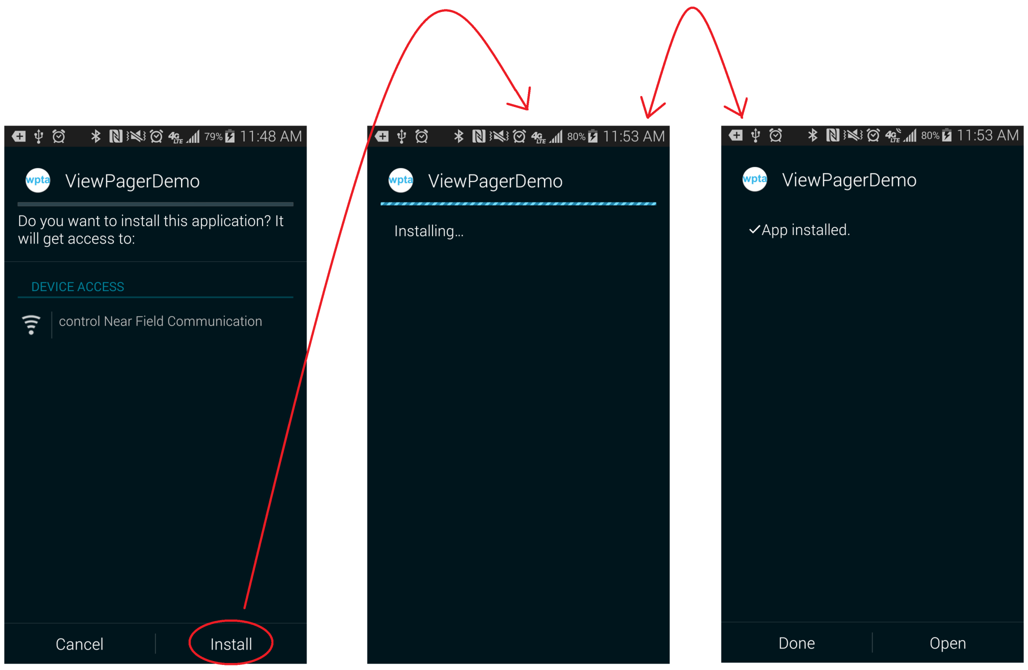


Figure 26. Successful App Installation

Although the app can run without enabling the NFC antenna, communication with the MPBSM patch cannot be established. The app has been configured to provide a warning pop-up menu should the app be launched without enabling the NFC antenna. Although the pop-up menu allows the user to continue without enabling the NFC radio, communication with the MPBSM EVM can never be established. Should the user choose to enable the NFC antenna, the pop-up menu provides a convenient shortcut to the NFC settings menu as shown in Figure 27.

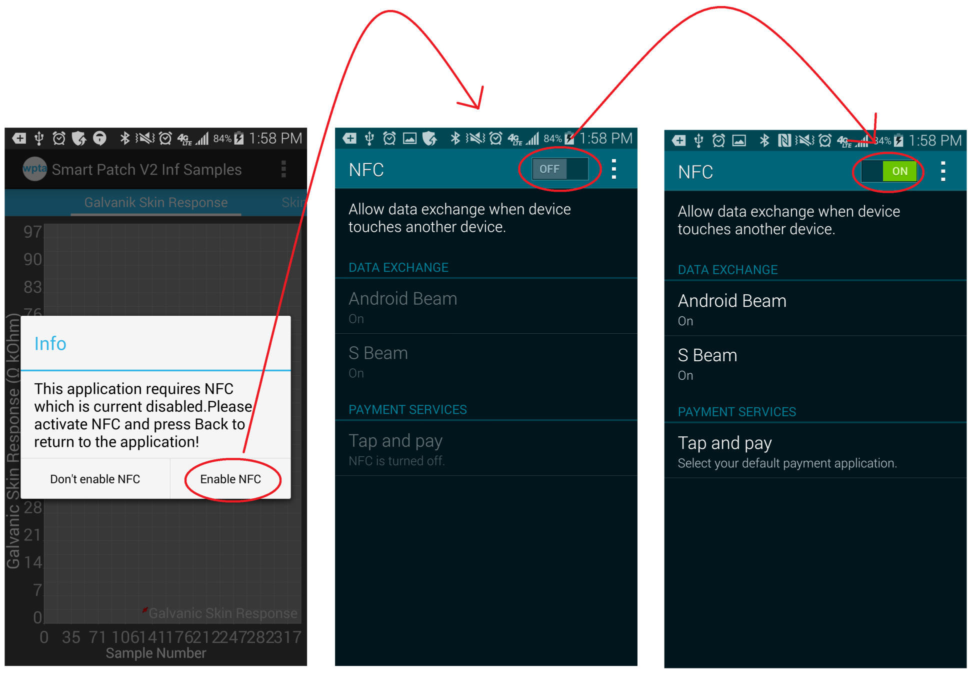

1:58 PM
Smart Patch V2 Inf Samples
Galvanik Skin Response
Info
This application requires NFC which is current disabled.Please activate NFC and press Back to return to the application!
Don't enable NFC
Enable NFC
Galvanic Skin Response (Ω kOhm)
Sample Number
NFC
OFF
Allow data exchange when device touches another device.
DATA EXCHANGE
Android Beam
On
S Beam
On
PAYMENT SERVICES
Tap and pay
NFC is turned off.
NFC
ON
Allow data exchange when device touches another device.
DATA EXCHANGE
Android Beam
On
S Beam
On
PAYMENT SERVICES
Tap and pay
Select your default payment application.

Figure 27. Enabling NFC Radio in Notifications Menu

5.3 Conductive Gel Application

Follow these steps to apply the conductive gel:

1. Apply conductive gel to the EVM probes to increase patch-to-skin contact and decrease noise. All the patch probes are located at the bottom of the patch.
2. Apply the supplied conductive gel to the EVM probes as shown in Figure 28.

A big problem with conductive gel is the smearing effect. When the patch's probes are topped with gel and placed on the patient's body, the conductive gel tends to smear in all directions. The smearing scenario is depicted in Figure 29. In Figure 29 a clear plexiglass is placed on bottom of the MPBSM to help illustrate the smearing effect.

By comparing Figure 28 to Figure 29, it can be seen how much smearing occurs. Excessive gel and excessive patch pressure leads to this gel smearing effect. If the smearing is sufficiently large, a short circuit can be created between the GSR and ECG probes, which would temporarily render the GSR and ECG readings incorrect. To minimize the chances of short circuiting, place gel on the opposite edges of the probes as shown in Figure 29. If gel happens to enter into the top of the patch, the entire MPBSM patch will be completely destroyed.

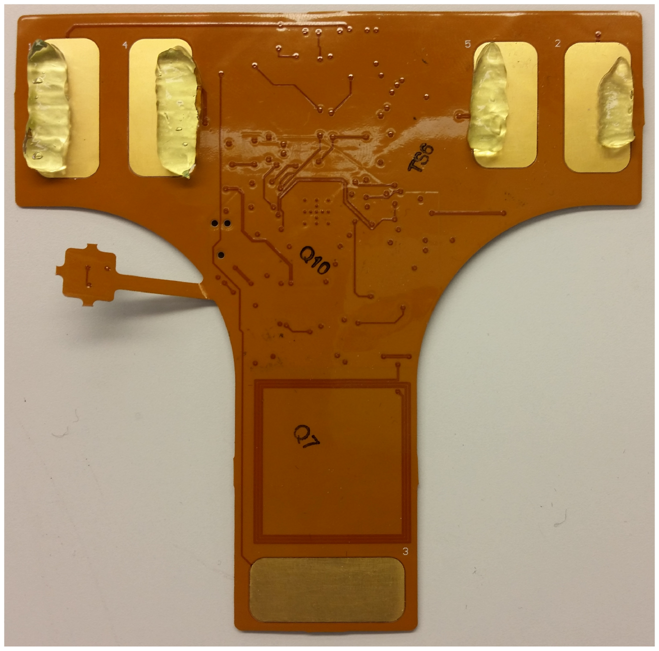


Figure 28. Conductive Gel Application

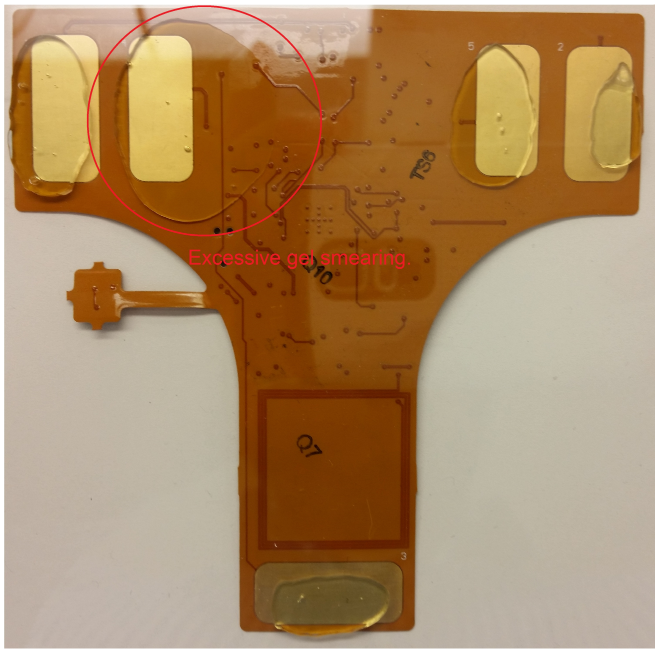


Figure 29. Smearing Effect

5.4 Tape Application

Tape is needed to adhere the MPBSM patch to the patient's skin. The type of tape (http://www.breacherstape.com/category-s/1822.htm) must be nonconductive and breathable with easy-to-remove adhesive. Breathable tape increases the level of comfort, and adhesive that easily peels off guarantees patch reuse. Figure 30 illustrates the tape application to the MPBSM system. As can be seen from Figure 30, the tape is placed on the top side of where the probes are located. This location increases patch-to-skin contact and decreases noise.

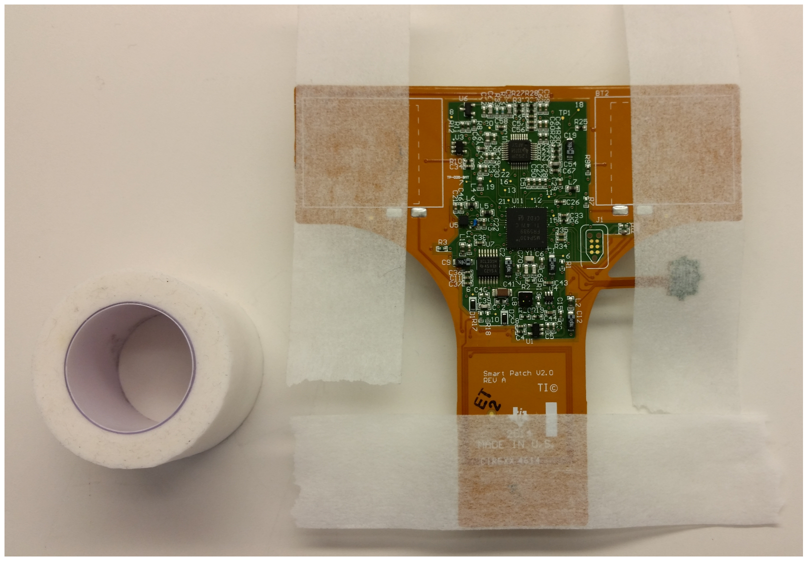


Figure 30. Smart Patch with Adhesive Tape

5.5 MPBSM Application

Place the MPBSM on the chest in such a way that the ECG leads are diagonally cross the heart. Placing the patch this way will provide the best possible ECG reading and heart-rate detection. The required patch position is illustrated in Figure 31.

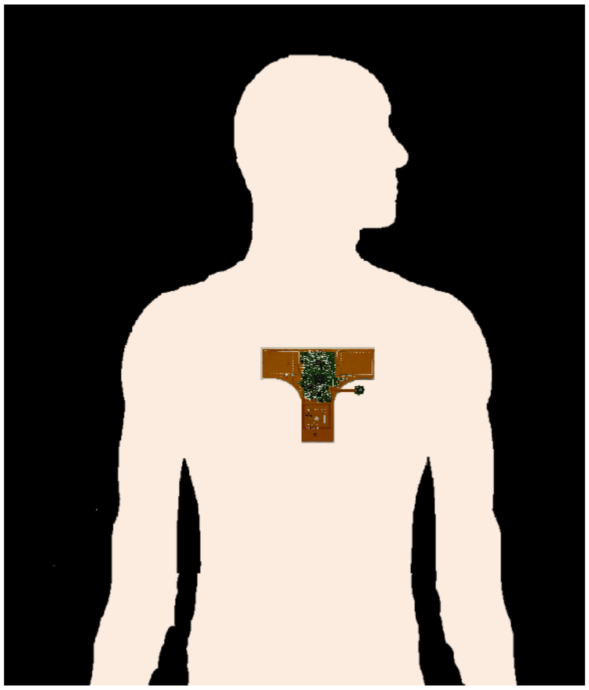


Figure 31. MPBSM Placement

5.6 MPBSM Scanning

Before attempting to scan the patch, familiarize yourself with the location of the NFC antenna on the phone and the antenna on the patch. The antenna on the patch is located on the top of the RLD ECG probe. The antenna is a 24 mm by 24 mm rectangle and is illustrated in Figure 32. The location of the NFC antenna on the phone is device dependent; however, a quick internet search should reveal its location. Locating the antennas should increase a pleasant user experience.

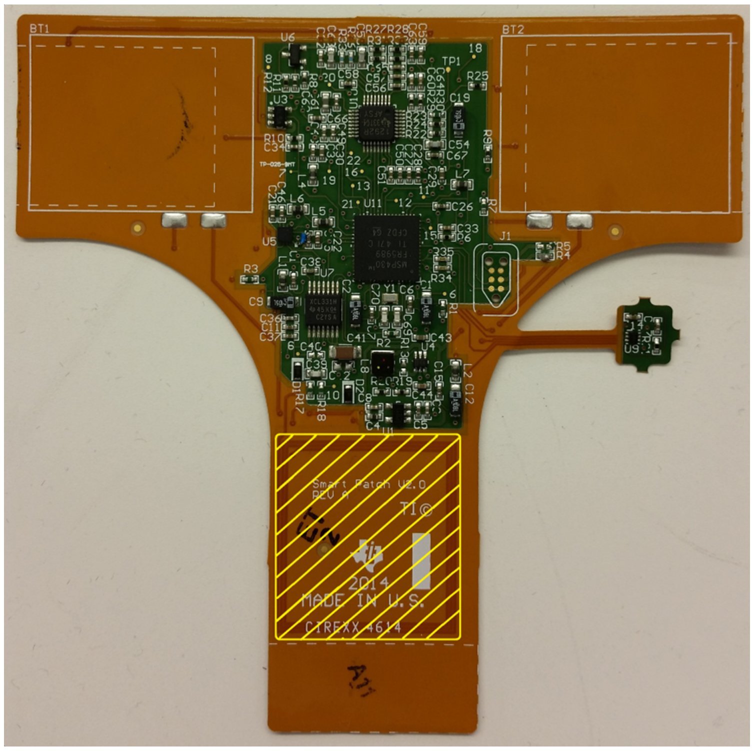


Figure 32. Antenna Location

To get the sensor readings, make sure the MPBSM app is in the foreground, then present the phone to the patch and wait until NFC ping sounds. A successful scan will make the NFC successful ping sound, and a “Scan Successful” toast message should appear at the bottom of the app. The “Scan Successful” toast message is illustrated in Figure 33. An unsuccessful scan will make an unsuccessful ping sound, and no toast message will appear at the bottom of the app. The difference between a successful and unsuccessful ping sound is given by the ping volume: a successful NFC ping will increase and decrease in volume, while an unsuccessful NFC ping only puts out a single volume ping. Once a successful scan is achieved, the graph should automatically update and reflect the sensor readings. At this point, try not to move the phone.

The MPBSM app has a variety of line graphs, each of which displays either sensor readings (ECG, GSR, or skin temperature) or the calculated readings (heart rate, respiration rate, cadence, step count). With the exception of the monolithic graph, each graph should only display one sensor's data. If another reading is desired, then the user can simply swipe until the desired sensor graph is shown, then re-scan the patch.

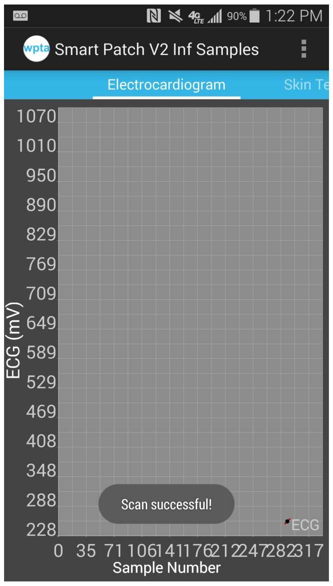


Figure 33. Successful Scan

6 Getting Started Firmware

The MPBSM software was written to maximize portability, modularity, and reusability. As a direct result, the software is broken up into separate independent software modules. Each module is either a driver module, an interface module, or an algorithm module. The MPBSM software uses eighteen software modules, of which seven are driver modules, seven are interface modules, and four are algorithm modules. The idea behind the MPBSM software came up from Linux device files. The MPBSM software modules are illustrated in Figure 34.

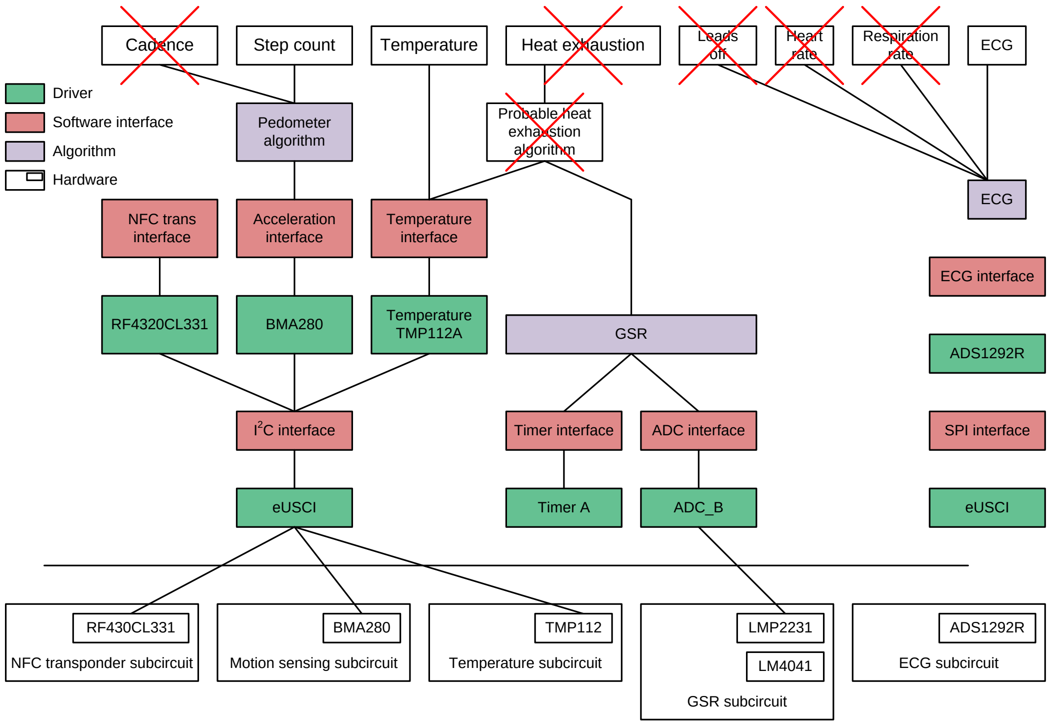


Figure 34. MPBSM Software Module Blocks

The driver modules directly interact with the external IC hardware and bridge the gap between hardware and software. In theory, the driver modules are the most likely to be replaced. A good example would be ADC enhancement. Suppose it is desired to replace the SAR 12-bit ADC for a Sigma-Delta-Converter driver module. The only modification needing to be done is the replacement of the SAR_12 driver modules. The driver module contains custom function calls and custom settings. Typically, but not always, the driver modules differ drastically from one module to another.

The driver modules work hand in hand with the interface modules, meaning that replacing a driver module requires altering its accompanied interface module. There are two types of interface modules: those that provide interface to MCU peripherals such as ADC_12, TimerB, eUSCI, and more; and those that provide interfaces to IC systems such as MPU9150 and BMA280. The reason behind this is because each function found in the interface module is full of driver-module-function calls. Module initialization, enableModule, and disableModule are some of the very common functions provided by almost all interface modules.

The algorithms modules are typically orders of magnitude bigger in size and complexity than either software interface modules' or driver modules' counterparts. The algorithm modules are most often math intensive and require a solid mathematical background. Most of the time, algorithm modules are created using desktop computer software such as MatLab.

Please be advised that the firmware documentation is considered the source code comments. The following sections will provide an overview of the most essential IC interface function. This function call names might change, but they will have a similar naming convention.

6.1 Temperature Interface Function List and Description

- TempInter _Init – This function initialized the temperature interface module.
- TempInter _PowerMode0 – This function call puts the TMP112 in active mode. This causes the IC to consume 10 µA.
- TempInter _ powerMode1 – This function call puts the TMP112 in shutdown mode. This causes the TMP112 to consume only 1 µA.
- TempInter _readTemperature – This function reads 2 bytes of temperature data from the TMP112.

6.2 Accelerometer Interface Function List and Description

- AcceleInter_ powerMode0 – This function causes the BMA280 to be in active mode. This causes the IC to consume 130 µA.
- AcceleInter_ powerMode1 – This function causes the BMA280 to be in LowPower mode 2. This causes the IC to consume 66 µA.
- AcceleInter_ powerMode2 – This function causes the BMA280 to be in LowPower mode 1. This causes the IC to consume 62 µA.
- AcceleInter_ powerMode3 – This function causes the BMA280 to be in Suspended mode. This causes the IC to consume 2.1 µA.
- AcceleInter_ powerMode4 – This function causes the BMA280 to be in Deep Suspended mode. This causes the IC to consume 1 µA.
- AcceleInter_readX – This function reads out the current X acceleration value (2 bytes).
- AcceleInter_readY – This function reads out the current Y acceleration value (2 bytes).
- AcceleInter_readZ – This function reads out the current Z acceleration value (2 bytes).

6.3 ECG Interface Function List and Description

- ADCInter_Init – This function initializes the ADS1292R module.
- ADCInter_PowerDown0 – This function causes the ADS1292R to be in active mode. This causes the IC to consume 670 µW.
- ADCInter_PowerDown1 – This function causes the BMA280 to be in active mode. This causes the IC to consume 160 µW.
- ADCInter_ReadECG – This function reads the current ADC conversion value.

6.4 NFC Transponder Interface Function List and Description

- NFCTranInter_Init – This function initializes the RF430CL331 module.
- NFCTranInter_PowerDown0 – This function causes the RF430CL331 to be in I2C mode. This causes the IC to consume 250 µA.
- NFCTranInter_PowerDown1 – This function causes the RF430CL331 to be in RF Enable mode. This causes the IC to consume 40 µA.
- NFCTranInter_PowerDown2 – This function causes the RF430CL331 to be in Inactive mode. This causes the IC to consume 15 µA.
- NFCTranInter_PowerDown2 – This function causes the RF430CL331 to be in Standby mode. This causes the IC to consume 10 µA.
- NFCTranInter_ReadECG – This function reads the current ADC conversion value.
- NFCTranInter_PowerDown1 – This function causes the eUSCI to be enabled.

6.5 Timer Interface Function List and Description

- NFCTranInter_Init – This function initializes the TimerA module.
- NFCTranInter_PowerDown0 – This function causes the TimerA to be disabled.
- NFCTranInter_PowerDown1 – This function causes the TimerA to be enabled.

6.6 NFCTransmitter Interface List and Description

- NFCTranInter_Init – This function initializes the ADC_B module.
- NFCTranInter_PowerDown0 – This function causes the ADC_B to be disabled.
- NFCTranInter_PowerDown1 – This function causes the ADC_B to be enabled.

6.7 Multiparameter Biosignal Monitor EVM Hardware Introduction

- NFCTranInter_Init – This function initializes the eUSCI module.
- NFCTranInter_PowerDown0 – This function causes the eUSCI to be disabled.
- NFCTranInter_PowerDown1 – This function causes the eUSCI to be enabled.

6.8 Pedometer Algorithm

The pedometer algorithm detects both the step count and cadence (steps per minute). As can be seen from Figure 35, the pedometer algorithm uses the acceleration values gathered by the BMA280 motion sensor to detect steps. All three acceleration axis values are used in the algorithm to calculate the step count. Furthermore, having all 3 axis acceleration data enables the step detection regardless of IC initial or final orientation. The pedometer algorithm requires the ADC to be running at a 62.5-Hz sampling rate. Slight deviation of the sampling frequency will result in incorrect step detection. The pedometer algorithm also requires the accelerometer to be set to 2G sensitivity and is provided as an executable library only. The source code for the pedometer algorithm is not provided; however, a pedometer.h file is provided. In the pedometer.h file, three functions are provided that can detect step count.

- ped_step_detect_init – This function initializes the pedometer algorithm module.
- ped_update_sample – The raw g values measured from the motion sensor are passed along to this function.
- ped_update_sample – This function determines if a step had been detected based on the G values passed to the ped_update_sample function.

6.9 NFC Commands

Commands given by the phone to the MPBSM are not single byte commands. Instead, a sequence of bytes determines what command was sent by the phone. To add to that complexity, every command is variable in size. As a consequence, a state machine is needed to process received bytes to determine if the received byte is the final byte in a command sequence or if it is another byte in the command sequence. The state machine used by the MPBSM is shown in Figure 35. If the byte received is the final byte in a command sequence, then the MPBSM should interpret the command.

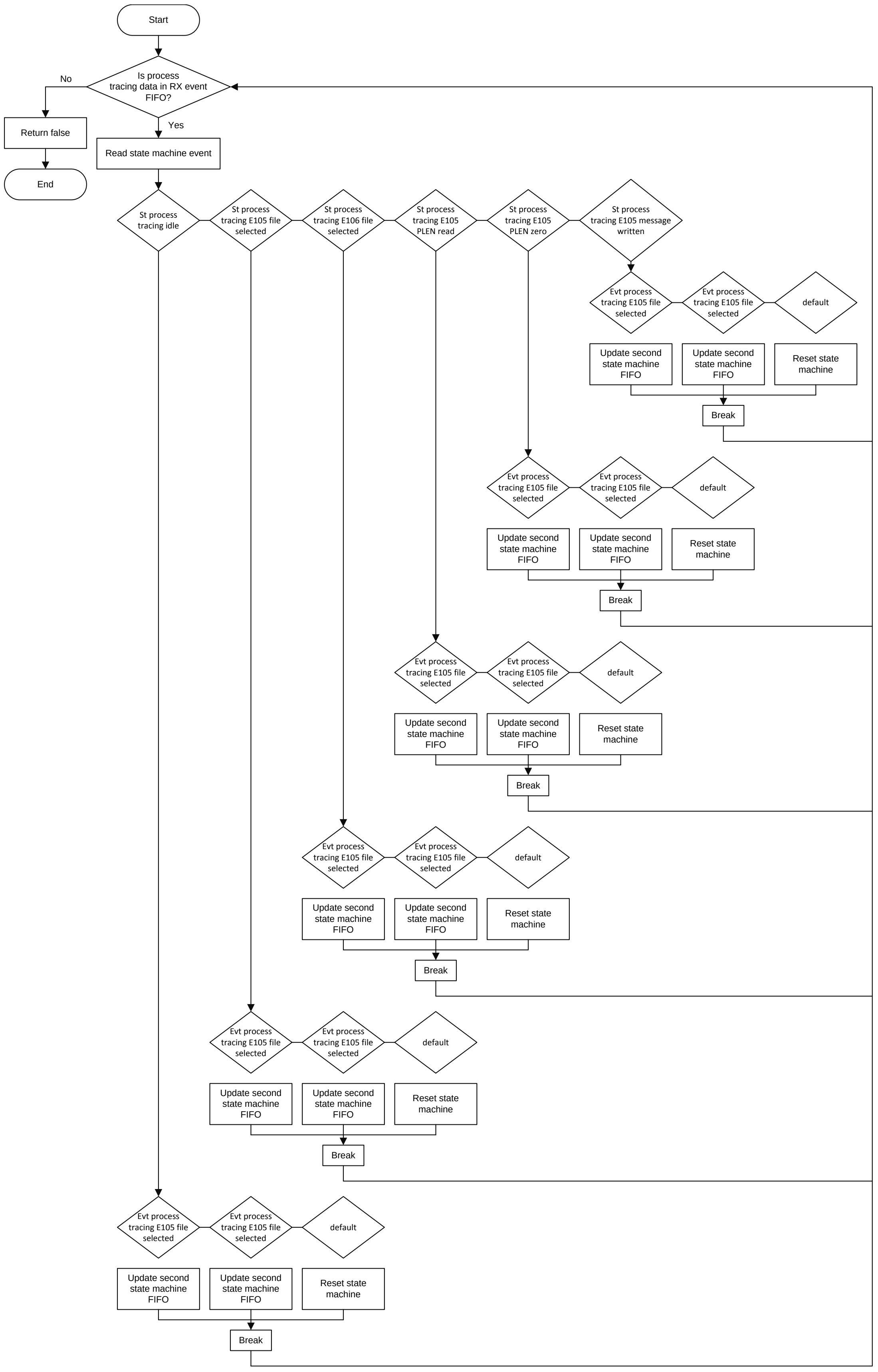

Start
Is process tracing data in RX event FIFO?
No
Yes
Return false
End
Read state machine event
St process tracing idle
St process tracing E105 file selected
St process tracing E106 file selected
St process tracing E105 PLEN read
St process tracing E105 PLEN zero
St process tracing E105 message written
Evt process tracing E105 file selected
default
Update second state machine FIFO
Reset state machine
Break

Figure 35. MPBSM Flow Chart

The cell phone and the MPBSM communicate by way of files. The Android phone is either reading to a file or writing to a file. The files are stored in the MPBSM and not on the Android phone. Although a copy of the file could be saved in the phone, the phone application never does this. Every time the phone reads a file, it simply stores the file in a temporarily buffer. The way the communication normally starts is the phone reads a special file called the FileTextE105 commands file. If the MPBSM is in the process of executing a command, then FileTextE105 will reflect that and hence the Android app must wait. Once the current command finishes executing and the MPBSM is free, then the Android app can send another command. Once every command finishes executing, both the results and its MPBSM-free status are written to the FileTextE105 command file.

7 MPBSM Android Application

Many features were added to the handset and MPBSM app to improve user experience (Zoom, Swipe, and reset zoom), increase applicability to a real world use case (Zoom, Time between samples, and Number of samples), increase usability (Disable swipe and swipe), and expand functionality (Time between samples and Number of samples). Unfortunately, with added capability comes added complexity. Although many of the features found in the MPBSM app were included to enhance the user experience, they may seem both intimidating and confusing. As a consequence, it was decided to dedicate a whole chapter purely to describing the MPBSM app, its features, and how they relate to the MPBSM system hardware.

The handset or MPBSM app extracts the sensor values using one of two methods: streaming or data-logging mode. Some features are available for one mode but not available for the other mode. For example, the “Number of samples” is not available in streaming mode. The advantage to streaming mode is that the signal values are observed in real-time and are gathered quickly and easily. The disadvantage in streaming is that the position of the handset must be held fixed for the entire duration of the scan. Conversely, for data-logger mode, the advantage is that the handset does not have to be held for the entire duration of sampling process. The disadvantage, however, for data-logger mode is that data values are not acquired as quickly because the MPBSM app must make a connection, grab the sample data, and finally close the connection instead of just making a connection and grabbing the sample data as streaming mode does. The next few paragraphs will describe each MPBSM app feature, which modes they are available in, and why they were added.

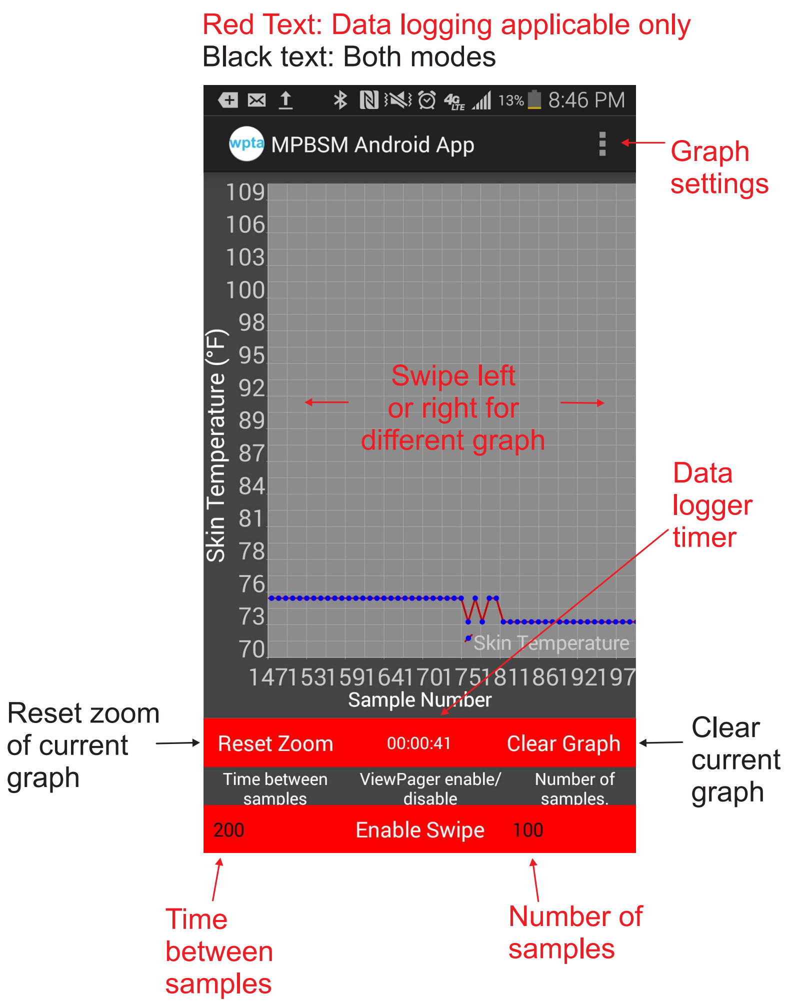
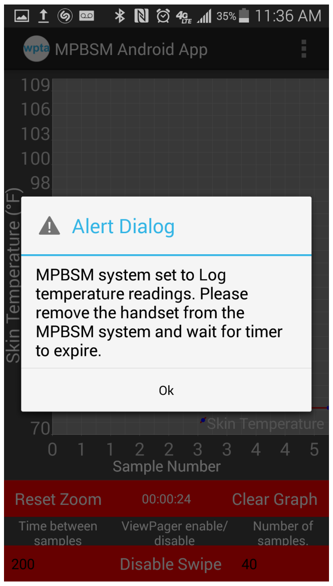


Figure 36. MPBSM App Features

7.1 *Data-Logger Mode*

Data logging is the process whereby the MPBSM system records the sensor values instead of immediately transmitting them to the handset or MPBSM app. In data-logger mode, the user is not required to hold the handset for the entire duration of the scan. Instead, in data-logger mode, the user scans the system only twice: initially to configure it and finally to read the recorded data values. The amount of time the MPBSM system spends collecting the data values is dependent on both the “time between samples” and the “number of samples” variables. Alert dialogs inform the user when and if the MPBMS system has been properly configured to data-logger mode and when the desired sensor values have been recorded and hence it's time to rescan the system to retrieve the sensor values.

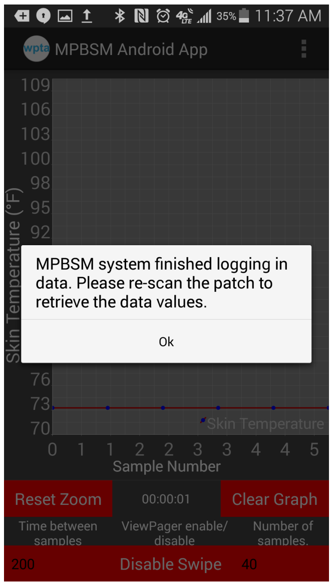
Both the successful-configuration dialog box and rescan-or-retrieve-sensor-values dialog box are illustrated in Figure 37 and Figure 38.


Alert Dialog

MPBSM system set to Log temperature readings. Please remove the handset from the MPBSM system and wait for timer to expire.

Ok

Figure 37. MPBSM System Configured for Data Logger Mode


MPBSM system finished logging in data. Please re-scan the patch to retrieve the data values.

Ok

Figure 38. Re-Scan MPBSM System

In addition to the alert dialogs, the handset will vibrate and notification pings will follow when both alert dialog boxes appear. While the MPBSM system is busy data logging sensor values, it should not be rescanned by the handset or Android app because it could potentially cause data corruption. As a preventive measure, the NFC transponder is disabled while the patch is data logging the sensor values. On the bottom center of the Android app is a countdown timer, which reveals how much time should elapse before a rescan should take place.

7.2 Streaming Mode

Streaming is the process whereby the MPBSM system transfers its sensor data values to the handset both continuously and in real-time. During streaming, the signal values are discarded as soon as they are transferred from the MPBSM system to the handset or Android app, meaning that the MPBSM ecosystem will continue to stream the data indefinitely as long as the handset or Android app is held on top of the system.

Just like data logger mode, streaming mode has its advantages and disadvantages. The advantage to streaming mode is that the signal values are observed in real-time and are gathered quickly and easily. The disadvantage to streaming is that the position of the handset must be held fixed for the entire duration of scan. Furthermore, during streaming mode, RF430CL331 IC must remain in active mode, which equates to higher current consumption. Streaming mode should be used if the time between samples is below 1 second and if the number of samples is below 100. To enable streaming mode, simply set the “time between samples” spinner box value and place the handset on top of the system as shown in Figure 31. A real-life application to streaming would be when a person other than the MPBSM wearer wishes to see body signals of the person wearing the MPBSM system. During streaming mode, the number of the sample variable is ignored.

7.3 Time Between Samples and Number of Samples Features

The time spent by the MPBSM system collecting data is dependent on both the “time between samples” and the “number of samples”. The image “Time between samples” determines how much time should elapse between the logging of sensor data. The value 300 is equal to approximately 1 second, the value 600 is equal to 2 seconds, and the value 900 is equal to 3 seconds. Any fractions of a second can also be determined by way of simple mathematics. The “number of samples” determines how many samples will be collected per session. For example, if the time between samples is set to 300 and the number of samples is set to 60, then the MPBSM system will be data logging for approximately 1 minute.

7.4 Zooming Feature

Whenever the number of sampled data points has exceeded 100, zooming is really appreciated. If a plot is overpopulated with data points, it becomes difficult to distinguish individual data-plot values. For that reason, the MPBSM Android app includes a plot-zooming feature, as shown in Figure 39.

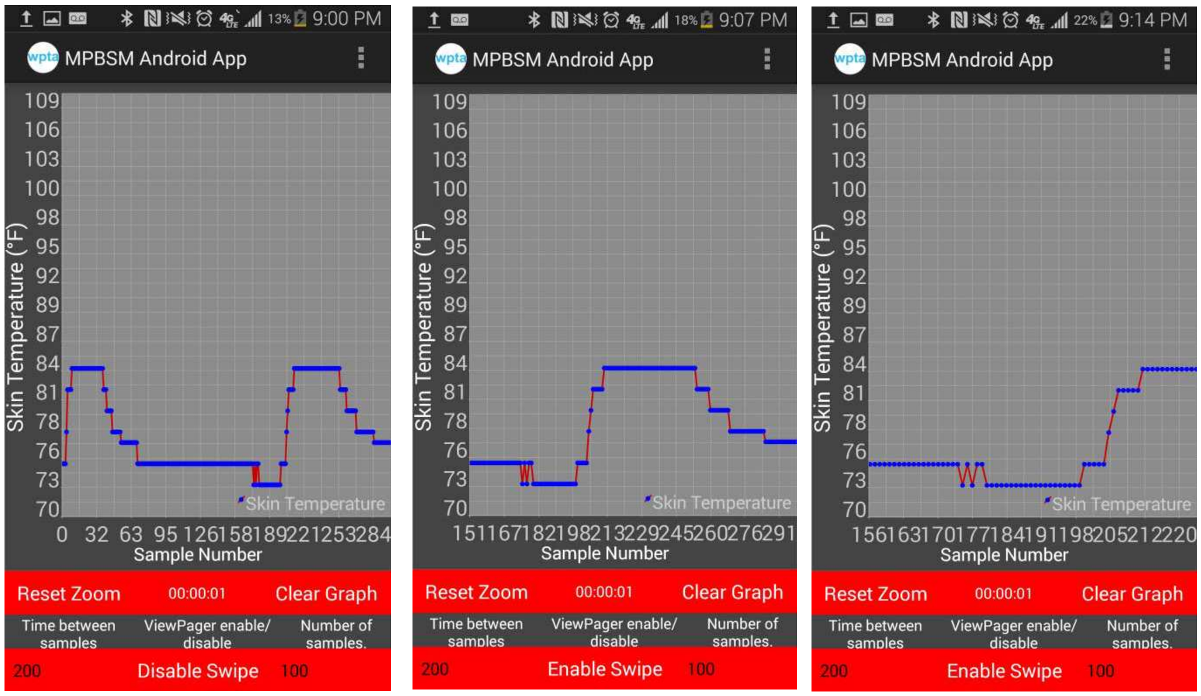


Figure 39. Swipe to Zoom in on Data Point Values

7.5 Swipe Feature

To create a more intuitive user interface, the swipe feature was incorporated into the Android app. Swiping in the Android app is used to transition from one plot to another. Furthermore, swiping is also used to see plotted data point values that are outside the plot window. The problem arises when the user swipes to see plotted data point values that are outside the plot window, but instead transitions to another plot. Conversely, the user could also swipe to transition from one plot to another but instead may swipe to see plotted data point values that are outside the plot window. To distinguish what the swipe is intended for, an additional disable-swipe widget has been added. The disable-swipe widget applies to the transitions from one plot to another. If the disable swipe widget is enabled, then swipe applies to plot transition. Conversely, if the disable-swipe widget is disabled, then the swipe applies to the plotted data point values that are outside the plot window.

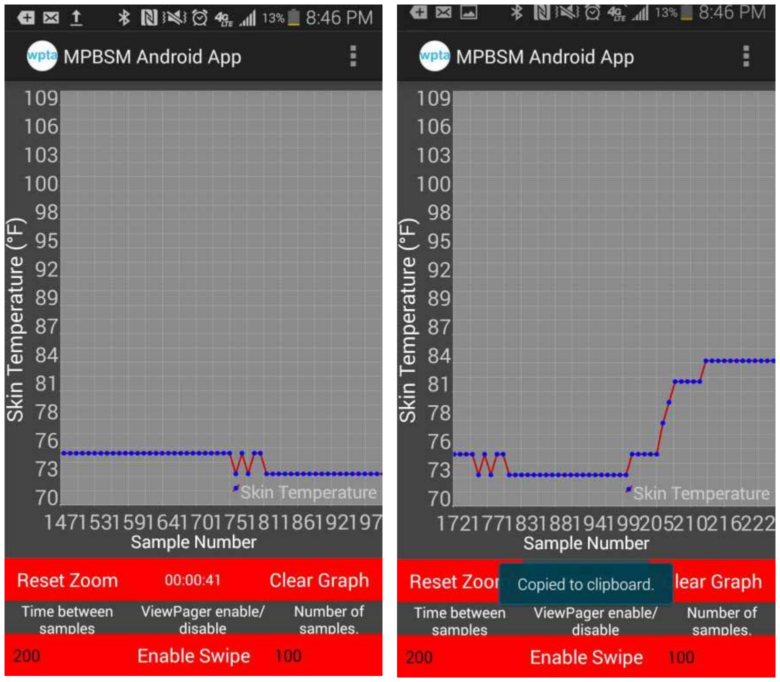


Figure 40. Swipe Left to See Data Point Values that are Outside the Plot Window

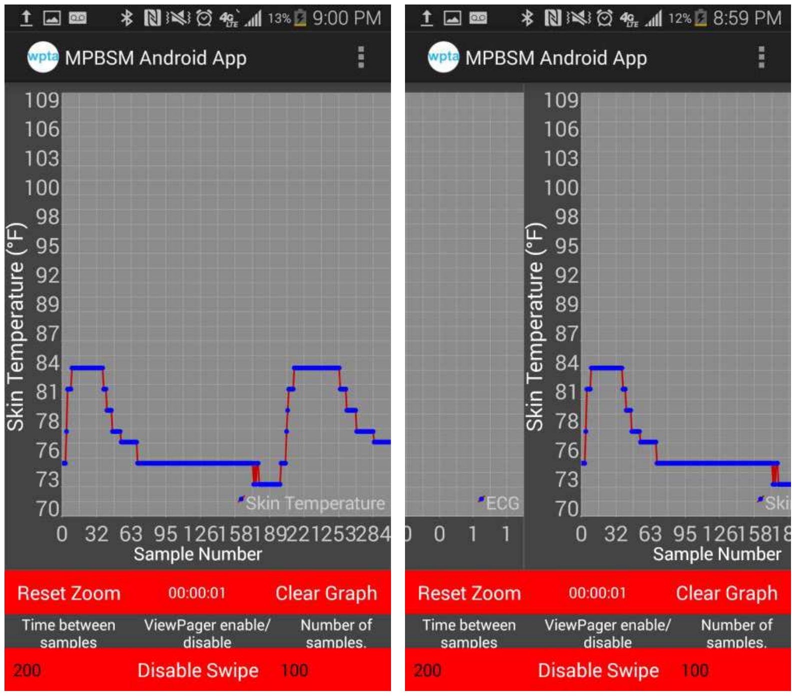


Figure 41. Swipe to Transition from One Plot to Another

7.6 Settings Menu

One unique thing about Android operating system is that it runs on a variety of devices from tablets, to laptops, to phones. Even within a specific range group, such as phones, there are a variety of hardware options such as screen size; even though the MPBSM Android app runs perfectly on an Android Galaxy S5 phone, it may not display correctly on an Android Galaxy S2 or S3 phone. To remedy this, a settings menu was incorporated into the device. As illustrated in Figure 41, to access the settings menu, simply click on the settings icon located in the upper right hand corner, and select the plot you wish to reconfigure. In the Android MPBSM app settings menu, you can find graph settings such as lower range boundaries, higher range boundaries, left margin, bottom margin, range step value, range ticks per label, range text size, range title text size, domain step value, domain ticks per label, domain text size, and domain title text size. It is recommended to experiment with the settings and see if the visual aspect of a graph can be improved.

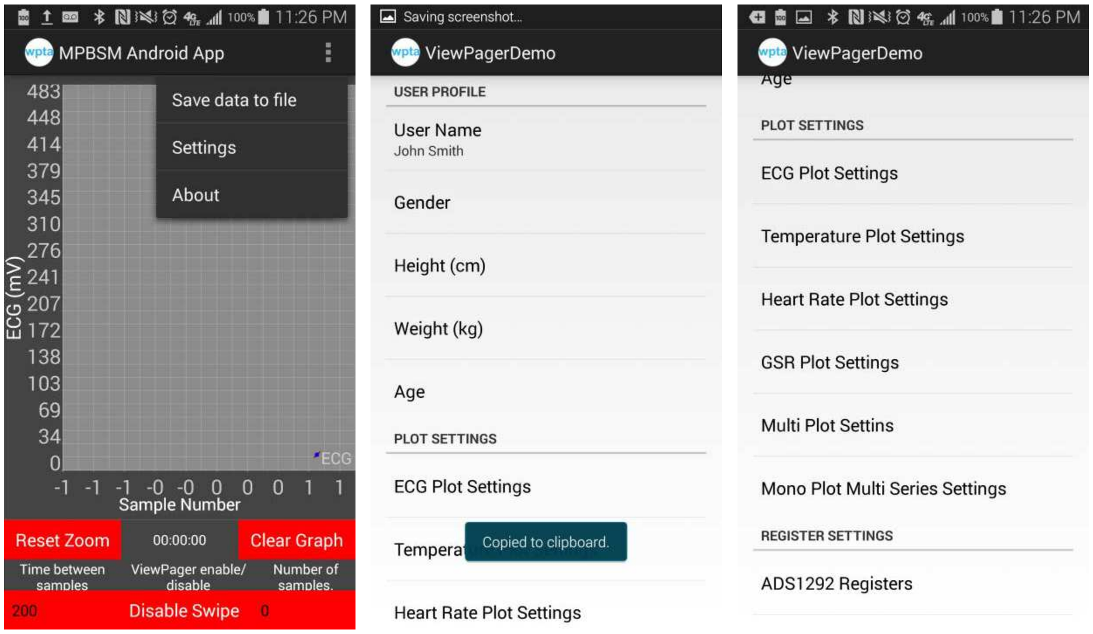
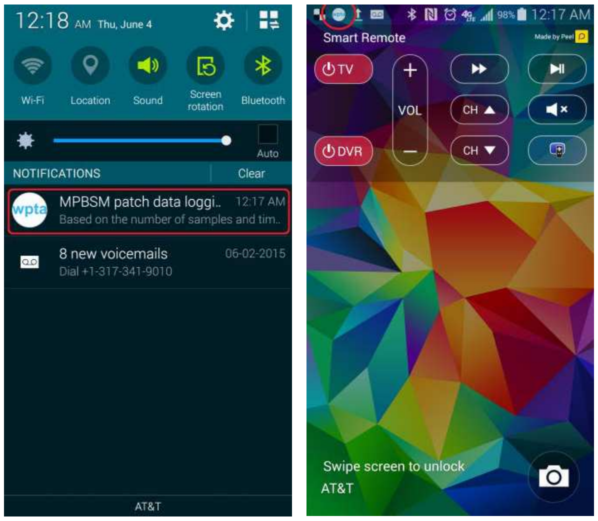


Figure 42. Android App Settings

7.7 Timer Alarm

As mentioned previously, the amount of time the MPBSM system spends collecting data is dependent on both the “time between samples” and the “number of samples”. This creates a problem because it is hard to determine when the system should be rescanned. A simple solution that is implemented is the incorporation of a stepdown counter. The starting time for the stepdown timer is computed using the “time between samples” and the “number of samples” variables and a constant value that is equivalent to the Timer_A clock frequency. What happens when the clock expires is dependent on whether the MPBSM app is in the foreground. If the MPBSM app is in the foreground, then the dialog box shown in Figure 38 is shown. If another application is on the foreground or if the handset is powered off, then the MPBSM app initiates a notification alert as shown in Figure 43.



MPBSM patch data loggi.. 12:17 AM
Based on the number of samples and tim..

8 new voicemails 06-02-2015
Dial +1-317-341-9010

Figure 43. MPBSM App Settings

7.8 Data Export

The data collected by the Android MPBSM app can be transferred to a text file. This process should help in algorithm development and debugging analysis. The final destination of the text file is in the app-install directory. The file created will have the TraceFile.txt name. Both the file name and destination patch are hardcoded.

8 Android App Source Description

This section is kept short due to heavy source code commenting. The MPBSM software is heavily commented and should be considered the documentation. In certain circumstances, the would-be comment is so long that it is not reasonable to place it in the source code. In those rare instances, the comment is placed here instead.

8.1 APDU_Commands.java

As was previously discussed, the NFC Android application communicates with the MPBSM by way of files. The commands and possible responses are described in this java file. As the APDU_Commands java file will attest, the Android app either selects a file, writes to a file, or reads from a file. None of the files are located on the Android device; they are merely stored in a temporary buffer while the application interprets their contents.

Some of the command responses give a variable length response. In those circumstances, the Android app dynamically allocates memory to temporarily store the file. All of the commands listed in APDU_Commands follow the Type 4 Tag Operation Specification. Reading the Type 4 Tag Operation Specification Technical Specification is a good starting point when trying to interpret the commands listed in APDU_Commands. The APDU_Commands java file also handles what happens when the expected response is not received. The user selects whether an error message is shown or if nothing happens. This feature is extremely helpful for debugging purposes; however, it becomes a nuisance elsewhere. One thing worth mentioning about the APDU_Commands is that all command responses (correct or incorrect) are routed to the LogCat terminal, which is very helpful during the debug phase.

8.2 Type4TagConstructAndSendCommands.java

In Android, long operations should not be performed in the UI thread. If a long operation is performed on the UI thread, the Android operating system thinks that an error occurred and hence tries to close the application. AsyncTask enables proper and easy use of the UI thread. This class allows performing background operations and publishing results on the UI thread without having to manipulate threads and / or handlers. An asynchronous task is defined by a computation that runs on a background thread and whose result is published on the UI thread.

Type4TagConstructAndSendCommand declares an AsyncTask class. This class is executed whenever the app communicates with the MPBSM system. The problem when any application must communicate with the outside world is knowing when (if ever) the outside world will respond. One example is if the phone is scanning the patch but is removed before the command response can be completed. If the scan function were to be implemented on the UI thread, the Android app would crash.

8.3 Plotter.java

If the MPBSM data is proven to be valid, it is plotted with the assistance of this file. All the functions needed to plot data along with plot settings are defined in this file. X and Y Range, X and Y resolution, X and Y starting and ending point, graph color, and number of graphs can all be changed with the use of the functions defined in this file. Unfortunately, Android does not support Plotting APIs. As a result, AndroidPlot graphing libraries were used.

9 Test Results

Current consumption helps determine how long a system will last with a given power source such as a battery. As can be imagined, the lower the power consumption, the longer the lifespan of the system. Although increasing the power source device will also increase lifespan, it will also increase both the physical size of the system and its overall weight. Current consumption can be determined both analytically and experimentally. Analytical current consumption is typically calculated during the design of the system, while experimental current consumption is measured once a system has been designed. By comparing the analytical versus the experimental current consumption, hardware designers have a reference point to determine how well the system was designed.

Three typical methods are used to determine experimental current consumption, and each have its pros and cons. Typically, experimental current consumption of any system is detected with an oscilloscope, a multi-meter, or a DC power analyzer. The multimeter for example, although it is both ubiquitous in most engineering laboratories and is cheap, will calculate the average current consumption over a period of time (1 second typically). An oscilloscope, on the other hand, is also ubiquitous and it calculates current consumption in real-time (with the help of a shunt resistor), but it is not cheap (some high-quality oscilloscopes are known to cost as much as a new car). A DC analyzer calculates current consumption in real-time, but it costs as much as a high-end oscilloscope and is not ubiquitous in most engineering laboratories. To determine the current consumption of the MPBSM system both faithfully and accurately, a Keysight n6705b DC power analyzer was used. The interesting aspect about the Keysight power analyzer is that it is both a power supply and a current analyzer.

The MPBSM system analytical current consumption is somewhat hard to calculate. Although some electrical components have a well-defined current consumption, others do not. Take, for example, the MSP430 IC. In the MPBSM system, the MSP430 is being clocked by the Digitally Controlled Oscillator (DCO). The MPBSM system sets the DCO to run at its maximum frequency, which is 8 MHz. According to the MSP430 datasheet, the MSP430 current consumption in active mode is dependent on the MSP430 core frequency. The formula for the MSP430 current consumption in active mode is 100 uAmps per MHz, which means that for the MPBSM system, whenever the MSP430 is active, it consumes 800 uAmps. Although it is easy to calculate and analyze the MSP430 active current consumption, it is not so easy to determine for instance the current consumption of the USCI peripheral. Although the USCI peripheral current consumption is negligible in standby mode, the active-mode current consumption is dependent on supply voltage, bus clock frequency, strength of the external pull-up, number of peers on the bus, bus parasitic capacitance, and more. Too many factors determine the USCI current consumption that not even a 'typical' value, much less a min / max bounding box, is possible.

The current consumption of the MPBSM system at any point in time is dynamic and depends on several factors: how many ICs are in active mode, which ICs are in active mode, and for how long the ICs are in active mode. Take, for example, temperature measurement versus cadence (step-count) detection. When in active mode, the TMP112A and BMA280 consume 130 uAmps and 10 uAmps respectively. Furthermore, cadence detection requires that the BMA280 sensor be in active mode all the time, while the temperature measure requires the IC to be enabled 13 msec before a temperature reading. Clearly, cadence calculation consumes more current than temperature measurement.

However, regardless of which mode the MPBSM system is in, it eventually requires the attention of the MSP430. As was previously mentioned, when in active mode, the MSP430 consumes 800 uAmps, meaning that it is recommended to place the MSP430 into standby mode whenever possible. In standby mode, the MSP430 consumes only 0.4 uAmps.

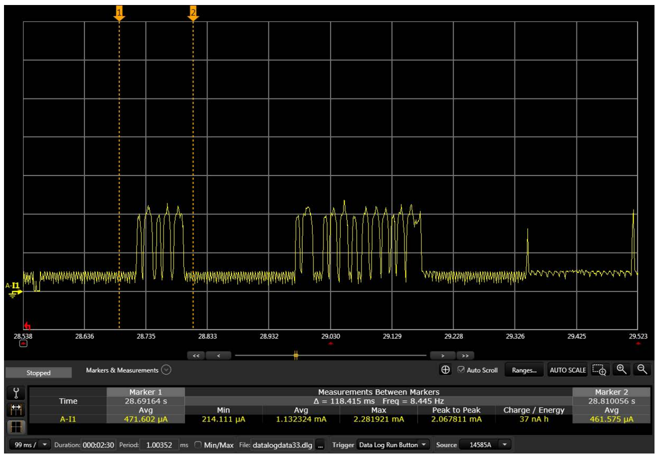
The following paragraphs describe in detail what the expected current consumption and measured current consumption of the BPBSM system is when set to a particular mode. A brief description as to how the calculated current consumption was calculated will also be given. The current analyze is broken down into three states; Initialization state, execution state, and power-down state. The current consumption of the ADS1292R, MSP430, TMP112A, BMA280, and RF430CL331 while in active mode is 340 uAmps, 800 uAmps, 10 uAmps, 130 uAmps, and 250 uAmps respectively. The current consumption of the ADS1292R, MSP430, TMP112A, BMA280 and RF430CL331 while in active mode is 20 uAmps, 0.4 uAmps, 1 uAmps, 62 uAmps, and 10 uAmps respectively. Regardless of which mode the MPBSM system is in, the USCI peripheral is always enabled.

9.1 Temperature Data Log Mode

Temperature data logger mode (TDLM) requires the TMP112A sensor, USCI (I^{2}C), and Timer_A peripherals to be in active mode. Furthermore, the RF430CL331 NFC transponder IC is in active mode coming to into, disabled while in, and finally re-activated when exiting TDLM mode. In addition, the MSP430 is partially in active mode.

9.1.1 TDLM Initialization State

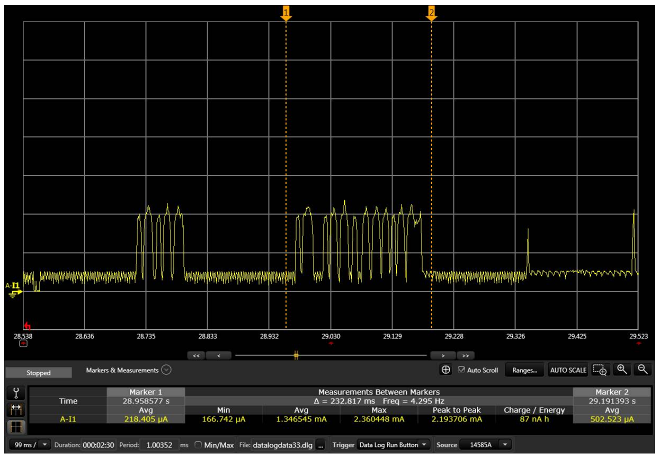
When entering into the TDLM Initialization state, the TMP112A, ADS1292R, BMA280, ADC12_B, and Timer_A are in standby mode while both the RF430CL331 and USCI are in active mode. During the TDLM Initialization state, both the TMP112A IC and the Timer_A peripheral are set to active mode while the RF430CL331 is set to standby mode.



	Marker 1	Measurements Between Markers					Marker 2
Time	28.69164 s	Δ = 118.415 ms Freq = 8.445 Hz					28.810056 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	471.602 µA	214.111 µA	1.132324 mA	2.281921 mA	2.067811 mA	37 nA h	461.575 µA

99 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata33.dlg Trigger Data Log Run Button Source 14585A

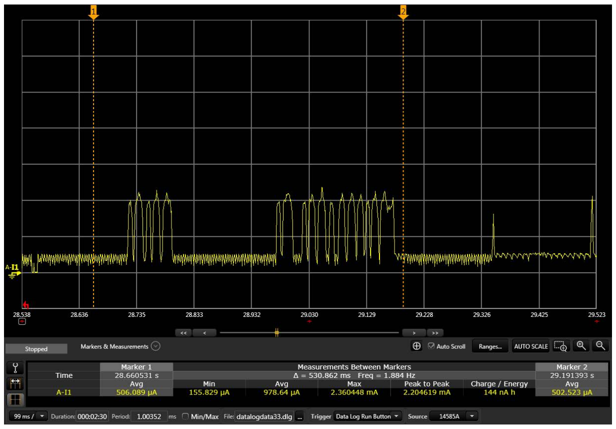
Figure 44. TDLM Initialization State—Timer_A Active



	Marker 1	Measurements Between Markers					Marker 2
Time	28.958577 s	Δ = 232.817 ms Freq = 4.295 Hz					29.191393 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	218.405 µA	166.742 µA	1.346545 mA	2.360448 mA	2.193706 mA	87 nA h	502.523 µA

99 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata33.dlg Trigger Data Log Run Button Source 14585A

Figure 45. TDLM Initialization State—TMP112A Active



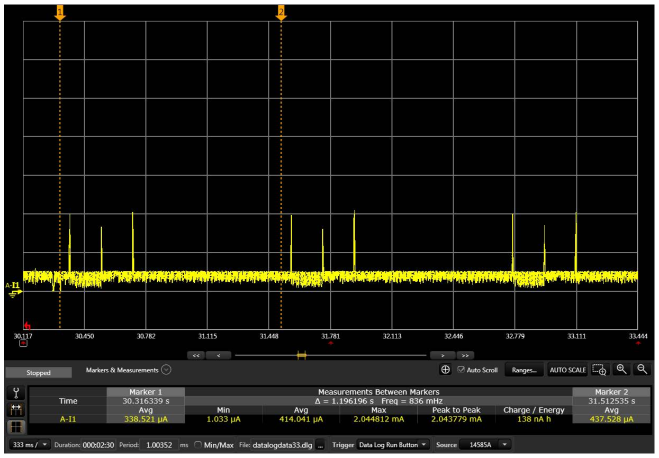
	Marker 1	Measurements Between Markers					Marker 2
Time	28.660531 s	Δ = 530.862 ms Freq = 1.884 Hz					29.191393 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	506.089 µA	155.829 µA	978.64 µA	2.360448 mA	2.204619 mA	144 nA h	502.523 µA

99 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata33.dlg Trigger Data Log Run Button Source 14585A

Figure 46. TDLM Initialization State—Timer_A and TMP112A Active

9.1.2 TDLM Execution State

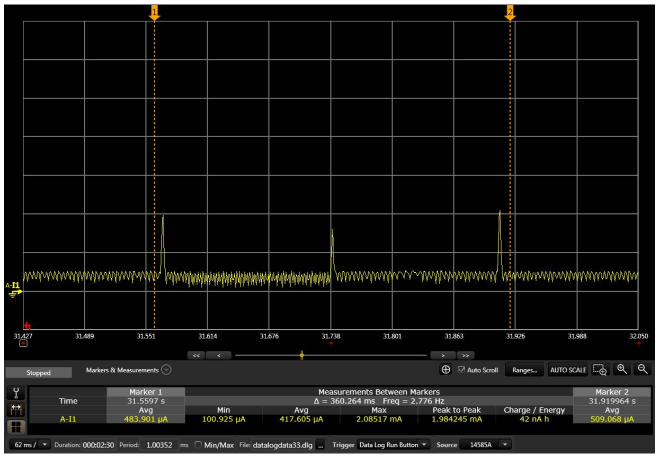
To reduce current consumption during the TDLM execution state, the TMP112A is repeatedly activated (active) and deactivated (standby). This is possible because the TMP112A requires only microseconds to send the temperature and requires milliseconds to settle when going from standby mode to active mode. The time between the temperature readings is dependent on the user inputs on the Android app. The Timer_A peripheral is in active mode during the entire duration of TDLM execution state. During the TDLM execution state, the analytical current consumption is 1180 uAmps plus Timer_A and USCI current.



	Marker 1	Measurements Between Markers					Marker 2
Time	30.316339 s	Δ = 1.196196 s Freq = 836 mHz					31.512535 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	338.521 µA	1.033 µA	414.041 µA	2.044812 mA	2.043779 mA	138 nA h	437.528 µA

333 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata33.dlg Trigger Data Log Run Button Source 14585A

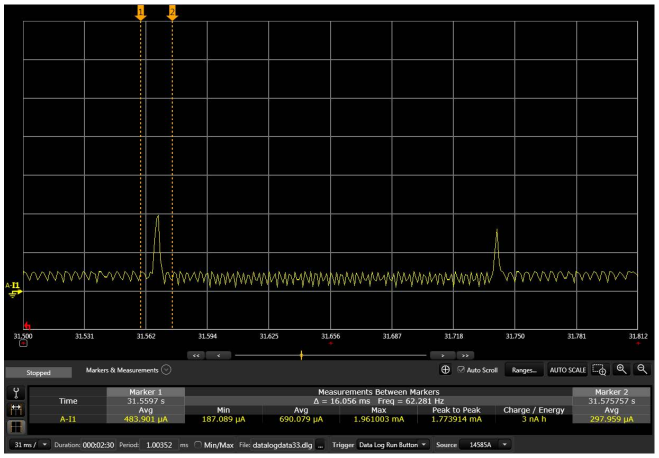
Figure 47. TDLM Execution State



	Marker 1	Measurements Between Markers					Marker 2
Time	31.5597 s	Δ = 360.264 ms Freq = 2.776 Hz					31.919964 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	483.901 µA	100.925 µA	417.605 µA	2.08517 mA	1.984245 mA	42 nA h	509.068 µA

62 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata33.dlg Trigger Data Log Run Button Source 14585A

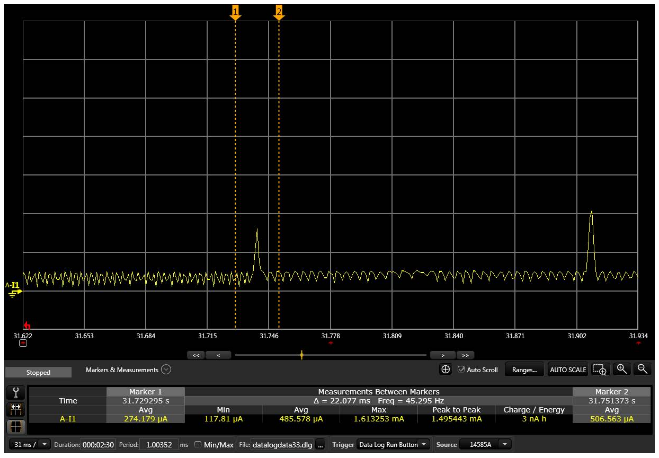
Figure 48. TDLM Execution State—Three-Stage Temperature Sample



	Marker 1	Measurements Between Markers					Marker 2
Time	31.5597 s	Δ = 16.056 ms Freq = 62.281 Hz					31.575757 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	483.901 µA	187.089 µA	690.079 µA	1.961003 mA	1.773914 mA	3 nA h	297.959 µA

31 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata33.dlg Trigger Data Log Run Button Source 14585A

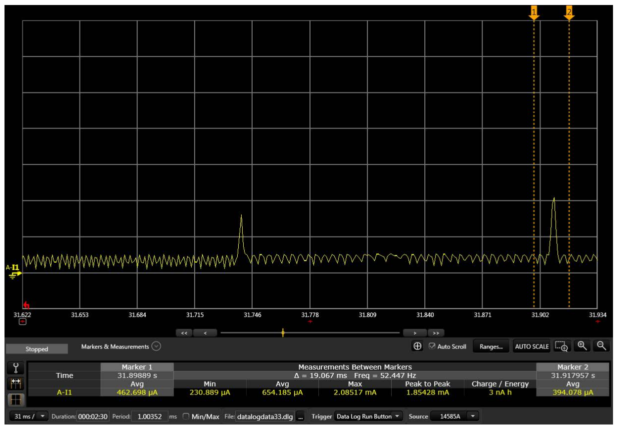
Figure 49. TDLM Execution State—Enable Timer and TMP112A Stage



	Marker 1	Measurements Between Markers					Marker 2
Time	31.729295 s	Δ = 22.077 ms Freq = 45.295 Hz					31.751373 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	274.179 µA	117.81 µA	485.578 µA	1.613253 mA	1.495443 mA	3 nA h	506.563 µA

31 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata33.dlg Trigger Data Log Run Button Source 14585A

Figure 50. TDLM Execution State—Read Temperature from TMP112A Stage



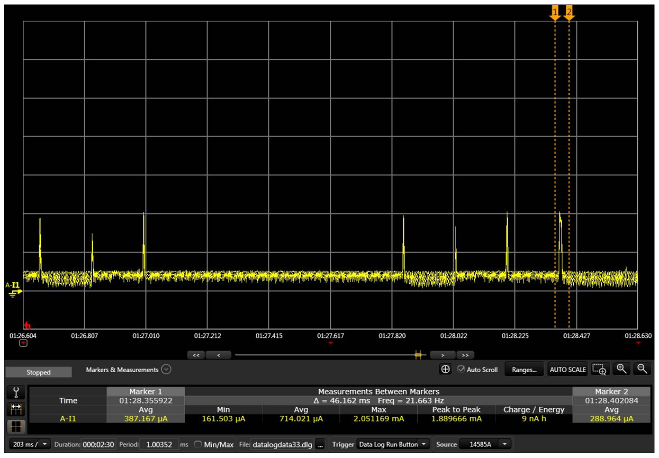
	Marker 1	Measurements Between Markers					Marker 2
Time	31.89889 s	Δ = 19.067 ms Freq = 52.447 Hz					31.917957 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	462.698 µA	230.889 µA	654.185 µA	2.08517 mA	1.85428 mA	3 nA h	394.078 µA

31 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata33.dlg Trigger Data Log Run Button Source 14585A

Figure 51. TDLM Execution State—Disable TMP112A Stage

9.1.3 TDLM Power Down State

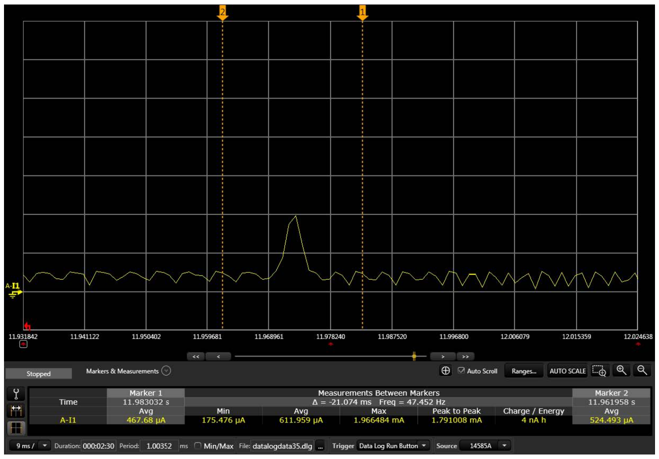
During TDLM power down, both the TMP112A and Timer_A are once again placed in standby mode.



	Marker 1	Measurements Between Markers					Marker 2
Time	01:28.355922	Δ = 46.162 ms Freq = 21.663 Hz					01:28.402084
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	387.167 µA	161.503 µA	714.021 µA	2.051169 mA	1.889666 mA	9 nA h	288.964 µA

203 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata33.dlg Trigger Data Log Run Button Source 14585A

Figure 52. TDLM Power Down State—Disable TMP112A Stage



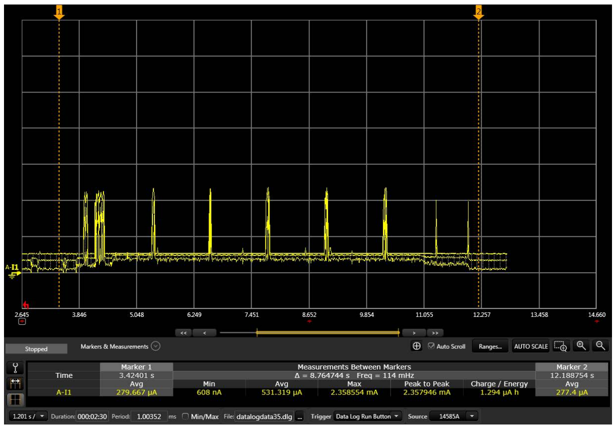
	Marker 1	Measurements Between Markers					Marker 2
Time	11.983032 s	Δ = -21.074 ms Freq = 47.452 Hz					11.961958 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	467.68 µA	175.476 µA	611.959 µA	1.966484 mA	1.791008 mA	4 nA h	524.493 µA

9 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata35.dlg Trigger Data Log Run Button Source 14585A

Figure 53. TDLM Power Down State—Zoom Disable TMP112A Stage

9.2 Temperature Streaming Mode

Temperature streaming mode (TSM) requires the TMP112A, RF430CL331, USCI (I^2C), and Timer_A to all be in active mode. Unlike TDLM mode, the RF430CL331 remains in active mode throughout the entire duration of the TSM mode. In addition, the MSP430 is partially in active mode.



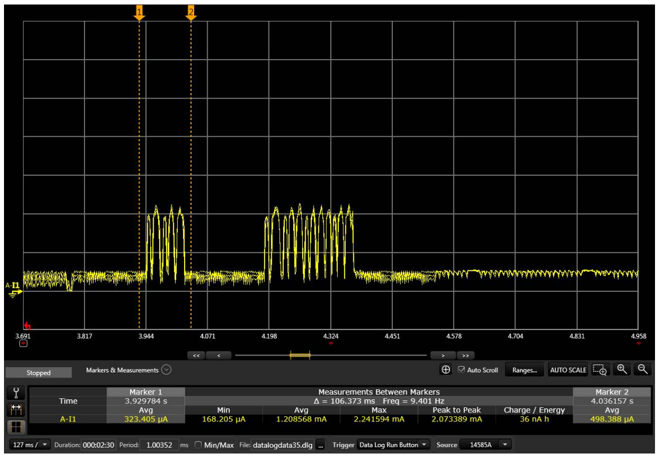
	Marker 1	Measurements Between Markers					Marker 2
Time	3.42401 s	Δ = 8.764744 s Freq = 114 mHz					12.188754 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	279.667 µA	608 nA	531.319 µA	2.358554 mA	2.357946 mA	1.294 µA h	277.4 µA

1.201 s / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata35.dlg Trigger Data Log Run Button Source 14585A

Figure 54. TSM 5 Samples

9.2.1 TSM Initialization State

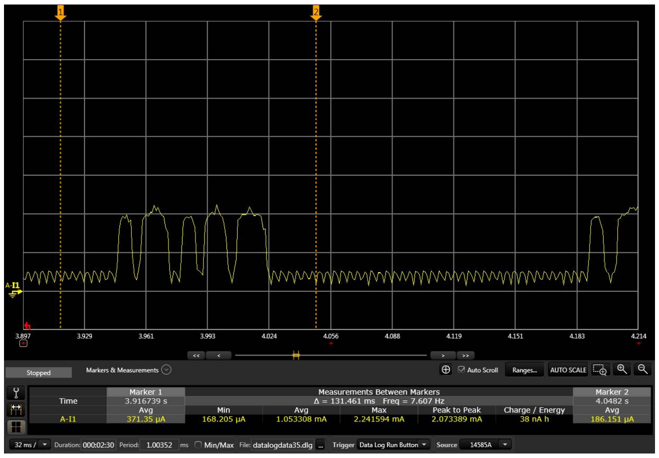
When entering into the TSM Initialization state, the TMP112A, ADS1292R, BMA280, ADC12_B, and Timer_A are in standby mode while both the RF430CL331 and USCI are in active mode. Unlike the TDLM Initialization state, the RF430CL331 is never placed in standby mode during the TSM Initialization state. During the TSM Initialization state, both the TMP112A IC and the Timer_A peripheral are set to active mode.



	Marker 1	Measurements Between Markers					Marker 2
Time	3.929784 s	Δ = 106.373 ms Freq = 9.401 Hz					4.036157 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	323.405 µA	168.205 µA	1.208568 mA	2.241594 mA	2.073389 mA	36 nA h	498.388 µA

127 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata35.dlg Trigger Data Log Run Button Source 14585A

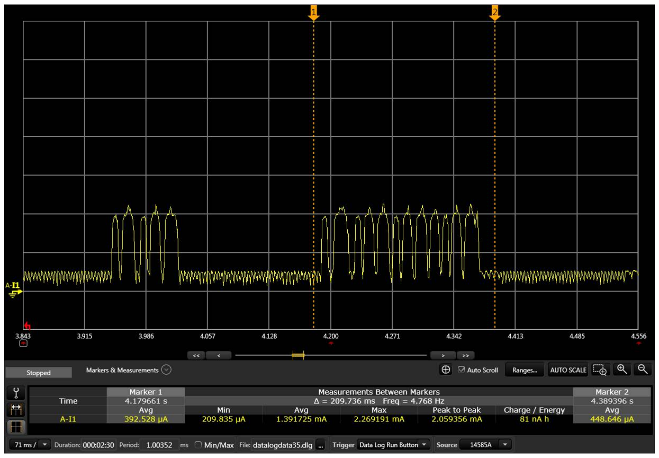
Figure 55. TSM Initialization State—TMP112A Active Stage



	Marker 1	Measurements Between Markers					Marker 2
Time	3.916739 s	Δ = 131.461 ms Freq = 7.607 Hz					4.0482 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	371.35 µA	168.205 µA	1.053308 mA	2.241594 mA	2.073389 mA	38 nA h	186.151 µA

32 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata35.dlg Trigger Data Log Run Button Source 14585A

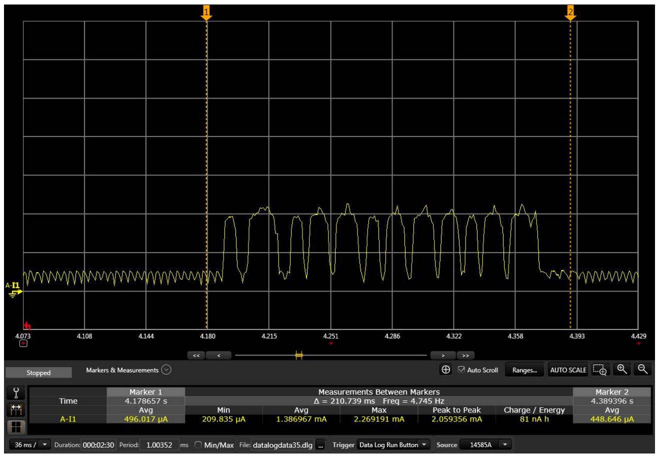
Figure 56. TSM Initialization State—TMP112A Active Stage Zoom



	Marker 1	Measurements Between Markers					Marker 2
Time	4.179661 s	Δ = 209.736 ms Freq = 4.768 Hz					4.389396 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	392.528 µA	209.835 µA	1.391725 mA	2.269191 mA	2.059356 mA	81 nA h	448.646 µA

71 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata35.dlg Trigger Data Log Run Button Source 14585A

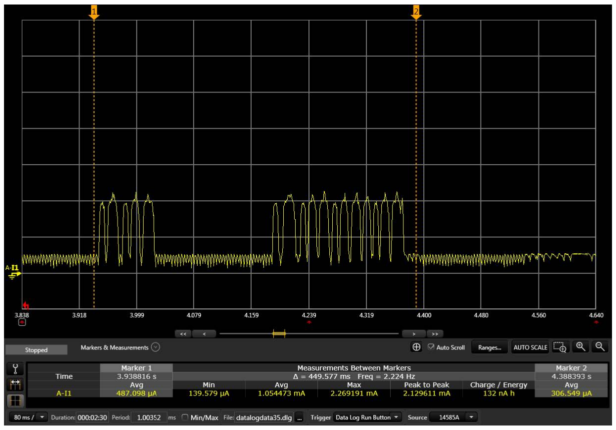
Figure 57. TSM Initialization State—Timer_A Active Stage



	Marker 1	Measurements Between Markers					Marker 2
Time	4.178657 s	Δ = 210.739 ms Freq = 4.745 Hz					4.389396 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	496.017 µA	209.835 µA	1.386967 mA	2.269191 mA	2.059356 mA	81 nA h	448.646 µA

36 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata35.dlg Trigger Data Log Run Button Source 14585A

Figure 58. TSM Initialization State—Timer_A Active Stage Zoom



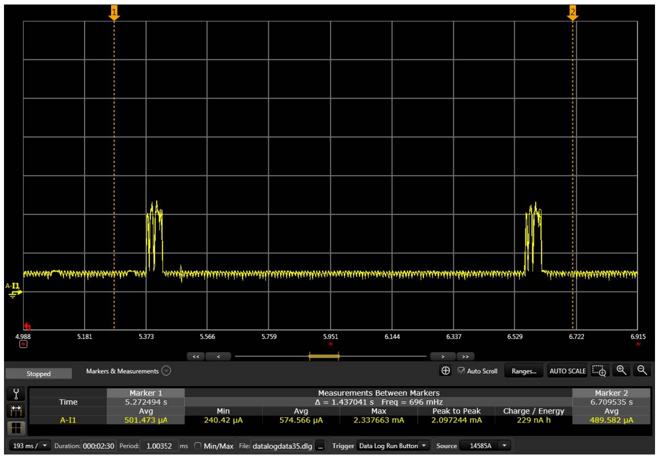
	Marker 1	Measurements Between Markers					Marker 2
Time	3.938816 s	Δ = 449.577 ms Freq = 2.224 Hz					4.388393 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	487.098 µA	139.579 µA	1.054473 mA	2.269191 mA	2.129611 mA	132 nA h	306.549 µA

80 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata35.dlg Trigger Data Log Run Button Source 14585A

Figure 59. TSM Initialization State—TMP112A and Timer_A Active

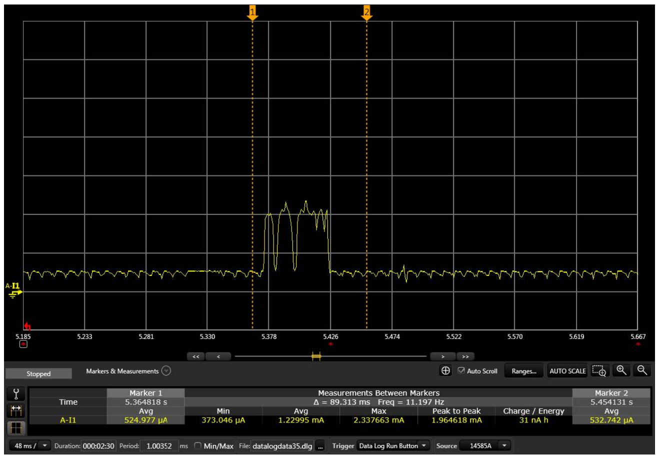
9.2.2 TSM Execution State

To reduce the possibility of data corruption and at the cost of power consumption, the TMP112A IC and RF430CL331 remain in active mode throughout the TSM execution state. Also, unlike the TDLM execution state, time between the temperature readings is fixed at one sample per second. As with TDLM execution state Timer_A peripheral is in active mode during the entire duration of the TSM execution state. During the TSM execution state, the analytical current consumption is 930 uAmps plus Timer_A and USCI current.



	Marker 1	Measurements Between Markers					Marker 2
Time	5.272494 s	Δ = 1.437041 s Freq = 696 mHz					6.709535 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	501.473 µA	240.42 µA	574.566 µA	2.337663 mA	2.097244 mA	229 nA h	489.582 µA

Figure 60. TSM Execution State—Two Samples

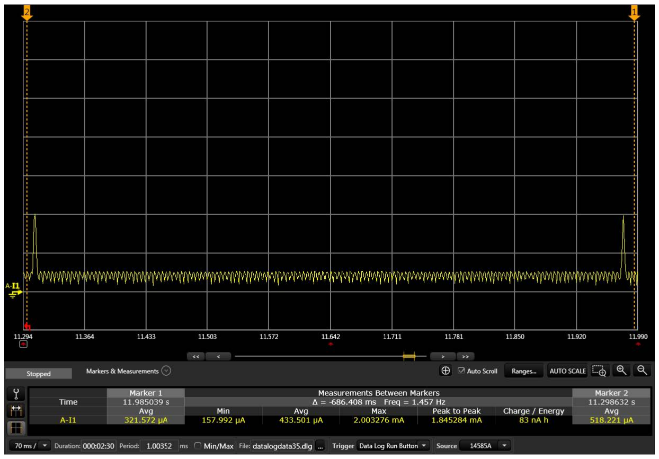


	Marker 1	Measurements Between Markers					Marker 2
Time	5.364818 s	Δ = 89.313 ms Freq = 11.197 Hz					5.454131 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	524.977 µA	373.046 µA	1.22995 mA	2.337663 mA	1.964618 mA	31 nA h	532.742 µA

Figure 61. TSM Execution State—First Sample

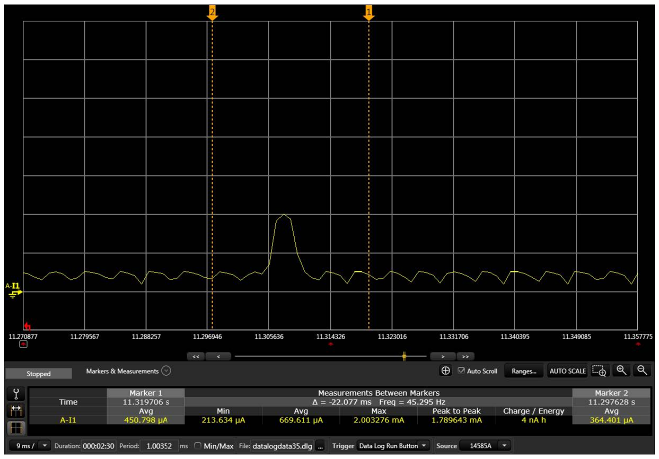
9.2.3 TSM Power Down State

During TSM power down, both the TMP112A and Timer_A are once again placed in standby mode. Also, unlike TDLM, the RF430CL331 is never set to active mode because it is already in active mode.



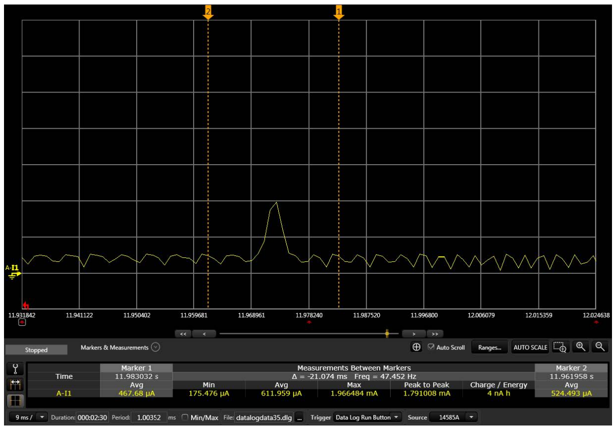
	Marker 1	Measurements Between Markers					Marker 2
Time	11.985039 s	Δ = -686.408 ms Freq = 1.457 Hz					11.298632 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	321.572 µA	157.992 µA	433.501 µA	2.003276 mA	1.845284 mA	83 nA h	518.221 µA

Figure 62. TSM Power Down State—Disable Timer_A and TMP112A Stage



	Marker 1	Measurements Between Markers					Marker 2
Time	11.319706 s	Δ = -22.077 ms Freq = 45.295 Hz					11.297628 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	450.798 µA	213.634 µA	669.611 µA	2.003276 mA	1.789643 mA	4 nA h	364.401 µA

Figure 63. TSM Power Down State—Disable Timer_A Stage



	Marker 1	Measurements Between Markers					Marker 2
Time	11.983032 s	Δ = -21.074 ms Freq = 47.452 Hz					11.961958 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	467.68 µA	175.476 µA	611.959 µA	1.966484 mA	1.791008 mA	4 nA h	524.493 µA

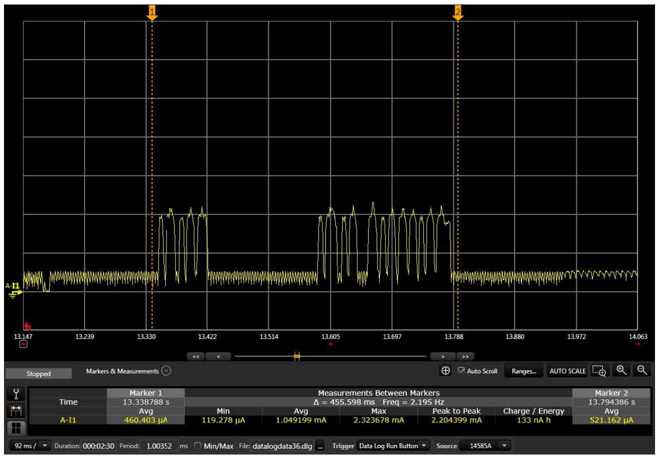
Figure 64. TSM Power Down State—Disable TMP112A Stage

9.3 GSR Data Log Mode

GSR data-log mode (GSRDLM) requires both the ADC12_B, Timer_A and the USCI (I^2C) peripherals to be in active mode. Furthermore, the RF430CL331 NFC transponder IC is in active mode coming into, disabled while in, and finally re-activated when exiting GSRDLM mode. In addition, the MSP430 is partially in active mode. As discussed before, the GSR subcircuit is a complete analog circuit and hence has no standby mode; it is always in active mode. The majority of the standby current consumption of the MPBSM system is attributed to the GSR subcircuit always being in the active mode.

9.3.1 GSRDLM Initialization State

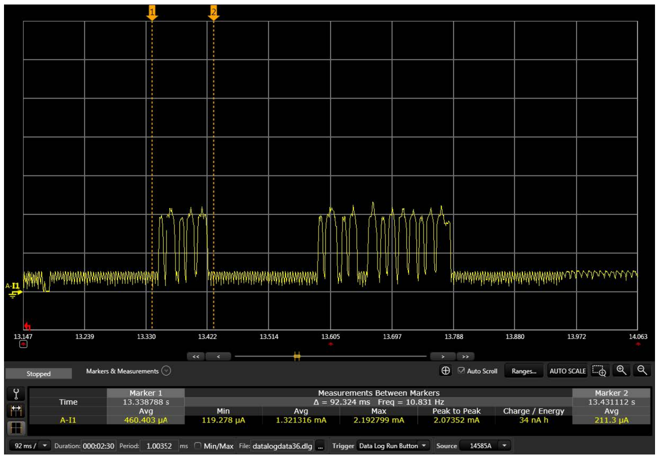
When entering into GSRDLM Initialization state, the TMP112A, ADS1292R, BMA280, ADC12_B, and Timer_A are in standby mode while both the RF430CL331 and USCI are in active mode. During the GSRDLM Initialization state, both the ADC12_B peripheral and the Timer_A peripheral are set to active mode while the RF430CL331 is set to standby mode.



	Marker 1	Measurements Between Markers					Marker 2
Time	13.338788 s	Δ = 455.598 ms Freq = 2.195 Hz					13.794386 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	460.403 µA	119.278 µA	1.049199 mA	2.323678 mA	2.204399 mA	133 nA h	521.162 µA

92 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata36.dlg Trigger Data Log Run Button Source 14585A

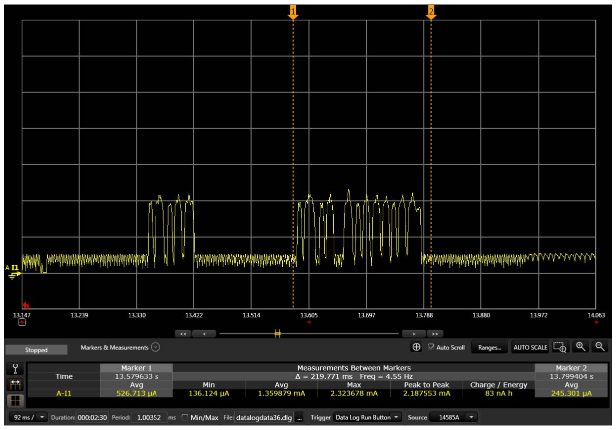
Figure 65. GSRDLM Initialization State—ADC12_B and Timer_A Active



	Marker 1	Measurements Between Markers					Marker 2
Time	13.338788 s	Δ = 92.324 ms Freq = 10.831 Hz					13.431112 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	460.403 µA	119.278 µA	1.321316 mA	2.192799 mA	2.07352 mA	34 nA h	211.3 µA

92 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata36.dlg Trigger Data Log Run Button Source 14585A

Figure 66. GSRDLM Initialization State—ADC12_B Active Stage



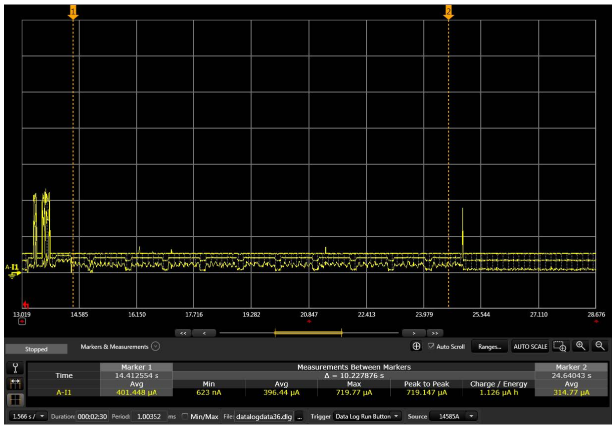
	Marker 1	Measurements Between Markers					Marker 2
Time	13.579633 s	Δ = 219.771 ms Freq = 4.55 Hz					13.799404 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	526.713 µA	136.124 µA	1.359879 mA	2.323678 mA	2.187553 mA	83 nA h	245.301 µA

92 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata36.dlg Trigger Data Log Run Button Source 14585A

Figure 67. GSRDLM Initialization State—Timer_A Active Stage

9.3.2 GSRDLM Execution State

Little effort is required by the MSP430 when in the GSRDLM execution state. The ADC12_B timer is sourced by the Timer_A peripheral, meaning that whenever a preconfigured Timer_A interrupt occurs, the ADC12_B samples the MSP430 ADC pin. Unfortunately, and as was discussed previously, the GSR subcircuit has no standby mode and hence is always in active mode regardless of whether the MPBSM system is in GSRDLM execution state or not. The time between the GSR readings is dependent on the user inputs on the Android app. The Timer_A peripheral is in active mode during the entire duration of the GSRDLM execution state. During GSRDLM execution state, the analytical current consumption is 920 uAmps plus Timer_A and USCI current.



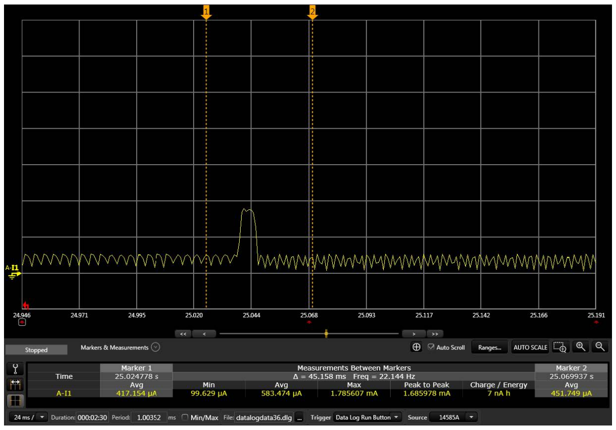
	Marker 1	Measurements Between Markers					Marker 2
Time	14.412554 s	Δ = 10.227876 s					24.64043 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	401.448 µA	623 nA	396.44 µA	719.77 µA	719.147 µA	1.126 µA h	314.77 µA

1.566 s / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata36.dlg Trigger Data Log Run Button Source 14585A

Figure 68. GSRDLM Execution State—Three Continuous Samples

9.3.3 GSRDLM Power Down State

During GSRDLM power down, both the ADC12_B and Timer_A are once again placed in standby mode. The current consumption save is considered negligible.



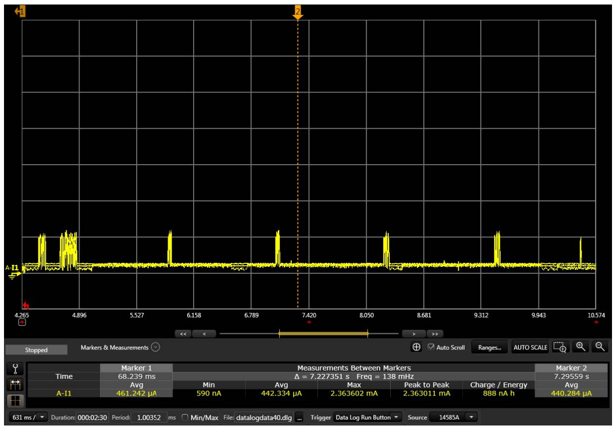
	Marker 1	Measurements Between Markers					Marker 2
Time	25.024778 s	Δ = 45.158 ms Freq = 22.144 Hz					25.069937 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	417.154 µA	99.629 µA	583.474 µA	1.785607 mA	1.685978 mA	7 nA h	451.749 µA

24 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata36.dlg Trigger Data Log Run Button Source 14585A

Figure 69. GSRDLM Power Down State—Disabled ADC Stage

9.4 GSR Streaming Mode

GSR streaming mode (GSRSM) requires both the ADC12_B, Timer_A, and the USCI (I^2C) peripherals to be in active mode. Unlike GSRDLM mode, the RF430CL331 remains in active mode throughout the entire duration of GSRDLM mode. In addition, the MSP430 is partially in active mode. As was discussed before the GSR subcircuit is a complete analog circuit and hence has no standby mode; it is always in active mode. The majority of the standby current consumption of the MPBSM system is attributed to the GSR subcircuit always being in the active mode.



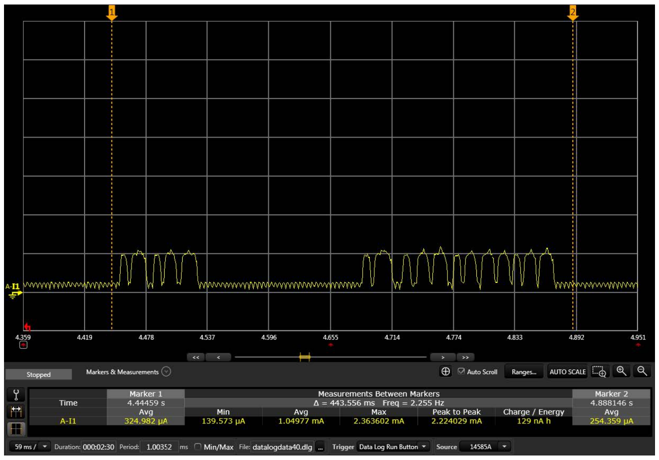
	Marker 1	Measurements Between Markers					Marker 2
Time	68.239 ms	Δ = 7.227351 s Freq = 138 mHz					7.29559 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	461.242 µA	590 nA	442.334 µA	2.363602 mA	2.363011 mA	888 nA h	440.284 µA

631 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata40.dlg Trigger Data Log Run Button Source 14585A

Figure 70. GSRSM Power Down State—Disabled ADC Stage

9.4.1 GSRSM Initialization State

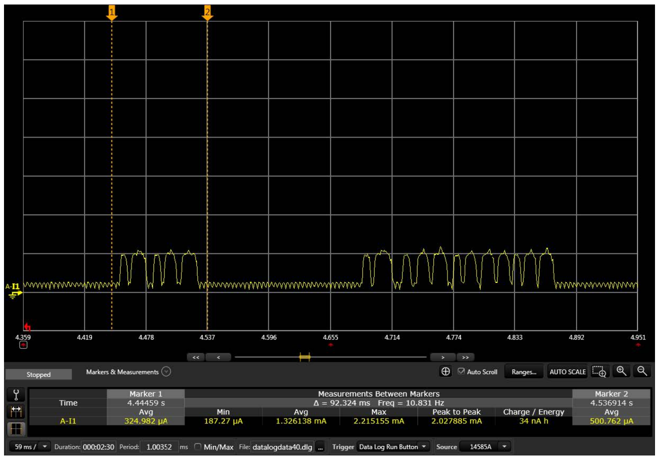
When entering into GSRSM initialization state, the TMP112A, ADS1292R, BMA280, ADC12_B, and Timer_A are in standby mode while both the RF430CL331 and USCI are in active mode. During the GSRDLM initialization state, the RF430CL331 is never placed in standby mode. During GSRSM initialization state, both the ADC12_B peripheral and the Timer_A peripheral are set to active.



	Marker 1	Measurements Between Markers					Marker 2
Time	4.44459 s	Δ = 443.556 ms Freq = 2.255 Hz					4.888146 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	324.982 µA	139.573 µA	1.04977 mA	2.363602 mA	2.224029 mA	129 nA h	254.359 µA

59 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata40.dlg Trigger Data Log Run Button Source 14585A

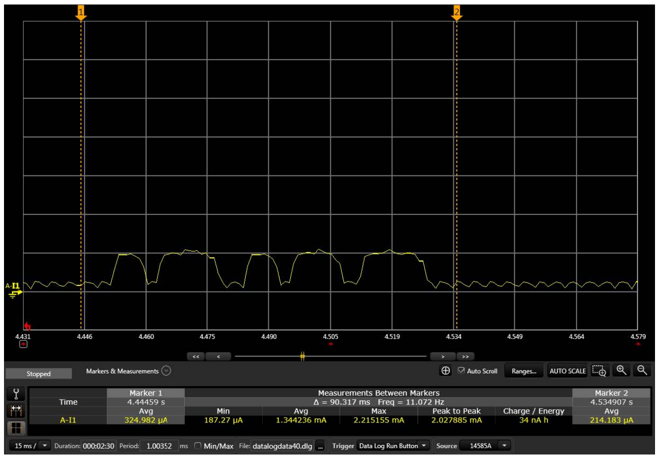
Figure 71. GSRSM Initialization State—ADC12_B and Timer_A Active



	Marker 1	Measurements Between Markers					Marker 2
Time	4.44459 s	Δ = 92.324 ms Freq = 10.831 Hz					4.536914 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	324.982 µA	187.27 µA	1.326138 mA	2.215155 mA	2.027885 mA	34 nA h	500.762 µA

59 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata40.dlg Trigger Data Log Run Button Source 14585A

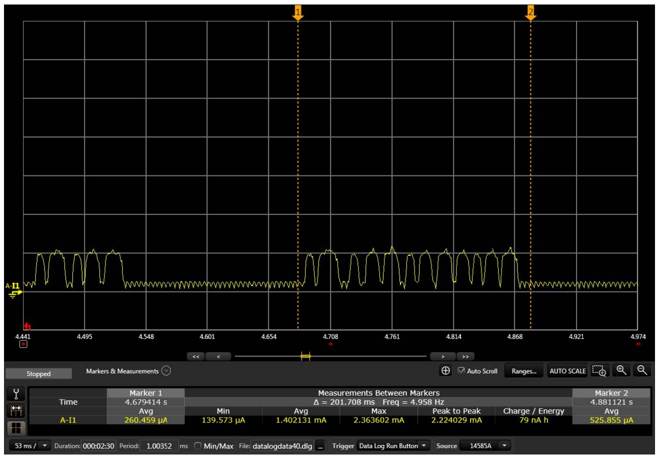
Figure 72. GSRSM Initialization State—ADC12_B Active Stage



	Marker 1	Measurements Between Markers					Marker 2
Time	4.44459 s	Δ = 90.317 ms Freq = 11.072 Hz					4.534907 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	324.982 µA	187.27 µA	1.344236 mA	2.215155 mA	2.027885 mA	34 nA h	214.183 µA

15 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata40.dlg Trigger Data Log Run Button Source 14585A

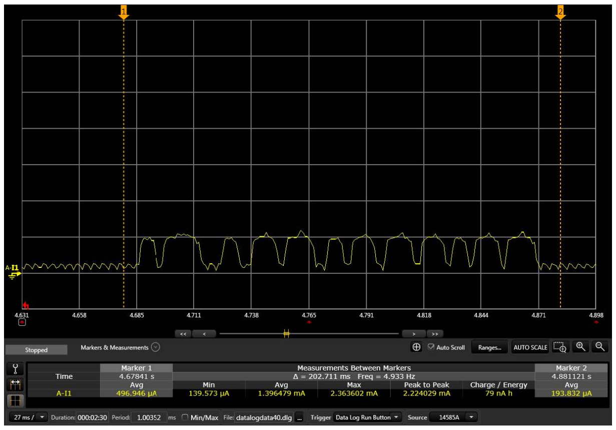
Figure 73. GSRSM Initialization State—ADC12_B Active Stage Zoom



	Marker 1	Measurements Between Markers					Marker 2
Time	4.679414 s	Δ = 201.708 ms Freq = 4.958 Hz					4.881121 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	260.459 µA	139.573 µA	1.402131 mA	2.363602 mA	2.224029 mA	79 nA h	525.855 µA

53 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata40.dlg Trigger Data Log Run Button Source 14585A

Figure 74. GSRSM Initialization State—Timer_A Active Stage



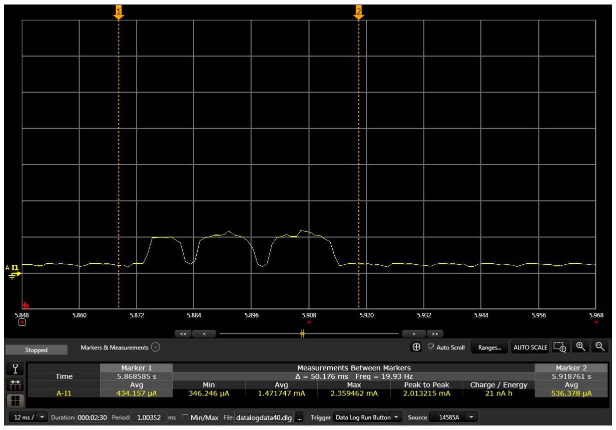
	Marker 1	Measurements Between Markers					Marker 2
Time	4.67841 s	Δ = 202.711 ms Freq = 4.933 Hz					4.881121 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	496.946 µA	139.573 µA	1.396479 mA	2.363602 mA	2.224029 mA	79 nA h	193.832 µA

27 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata40.dlg Trigger Data Log Run Button Source 14585A

Figure 75. GSRSM Initialization State—Timer_A Active Stage Zoom

9.4.2 GSRSM Execution State

Little effort is required by the MSP430 when in GSRSM execution state. The ADC12_B timer is sourced by the Timer_A peripheral, meaning that whenever a preconfigured Timer_A interrupt occurs, the ADC12_B samples the MSP430 ADC pin. Unfortunately, and as was discussed previously, the GSR subcircuit has no standby mode and is always in active mode regardless if the MPBSM system is in GSRSM execution state or not. In order to reduce the possibility of data corruption and at the cost of power consumption, the RF430CL331 remains in active mode throughout GSRSM execution state. Also, unlike GSRDLM execution state time between the temperature readings is fixed at one sample per second. The Timer_A peripheral is in active mode during the entire duration of GSRSM execution state. During GSRSM execution state, the analytical current consumption is 1170 uAmps plus Timer_A and USCI current.



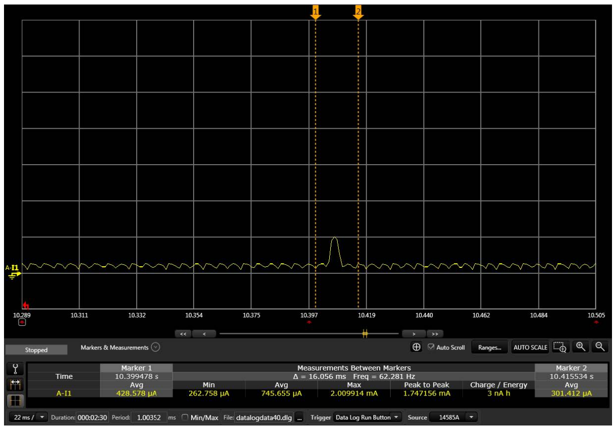
	Marker 1	Measurements Between Markers					Marker 2
Time	5.868585 s	Δ = 50.176 ms Freq = 19.93 Hz					5.918761 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	434.157 µA	346.246 µA	1.471747 mA	2.359462 mA	2.013215 mA	21 nA h	536.378 µA

12 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata40.dlg Trigger Data Log Run Button Source 14585A

Figure 76. GSRSM Execute State—One Continuous GSR Sample

9.4.3 GSRSM Power Down State

During GSRSM power down, both the ADC12_B and Timer_A are once again placed in standby mode. Also, unlike GSRDLM the RF430CL331 is never set to active mode during GSRSM power down because it is already in active mode.



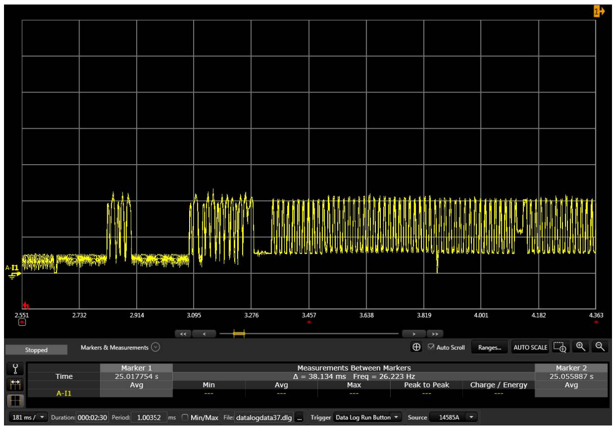
	Marker 1	Measurements Between Markers					Marker 2
Time	10.399478 s	Δ = 16.056 ms Freq = 62.281 Hz					10.415534 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	428.578 µA	262.758 µA	745.655 µA	2.009914 mA	1.747156 mA	3 nA h	301.412 µA

22 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata40.dlg Trigger Data Log Run Button Source 14585A

Figure 77. GSRSM Power Down State—Disabled ADC Stage

9.5 Cadence Data Log Mode

Cadence data log mode (CDLM) requires both the BMA280, Timer_A, and the USCI (I^2C) peripherals to be in active mode. Furthermore, the RF430CL331 NFC transponder IC is in active mode coming to into, disabled while in, and finally re-activated when exiting CDLM mode. In addition, the MSP430 is partially in active mode.



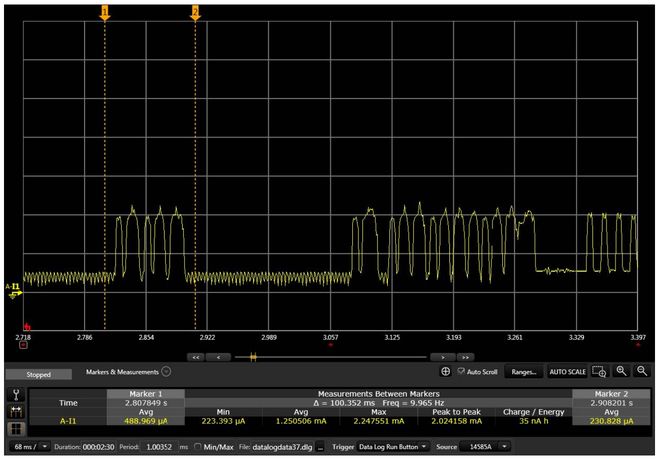
	Marker 1	Measurements Between Markers					Marker 2
Time	25.017754 s	Δ = 38.134 ms Freq = 26.223 Hz					25.055887 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1		---	---	---	---	---	

181 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata37.dlg Trigger Data Log Run Button Source 14585A

Figure 78. CDLM Initialization and Execution States

9.5.1 CDLM Initialization State

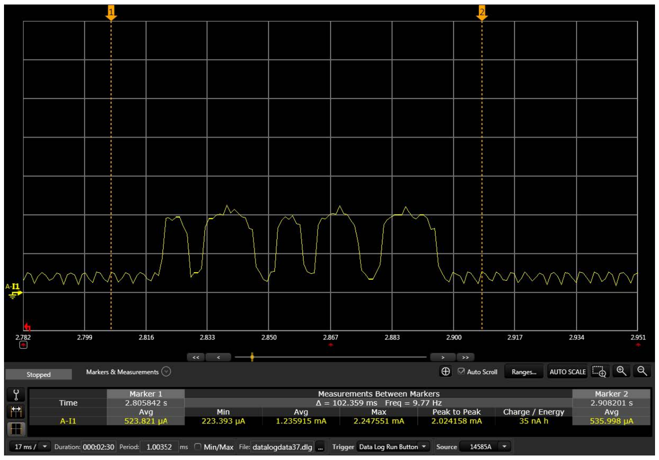
Entering into CDLM Initialization state the TMP112A, ADS1292R, BMA280, ADC12_B, and Timer_A are in standby mode while both the RF430CL331 and USCI are in active mode. During CDLM Initialization state both the BMA280 peripheral and the Timer_A peripheral are set to active mode while the RF430CL331 is set to standby mode.



	Marker 1	Measurements Between Markers					Marker 2
Time	2.807849 s	Δ = 100.352 ms Freq = 9.965 Hz					2.908201 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	488.969 µA	223.393 µA	1.250506 mA	2.247551 mA	2.024158 mA	35 nA h	230.828 µA

68 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata37.dlg Trigger Data Log Run Button Source 14585A

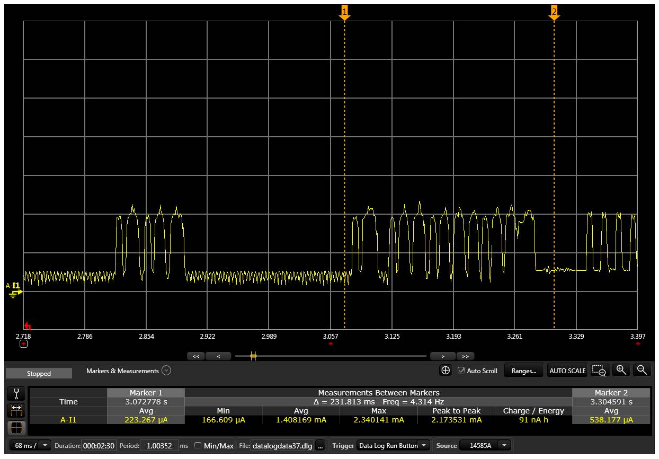
Figure 79. CDLM Initialization State—BMA280 Active Stage



	Marker 1	Measurements Between Markers					Marker 2
Time	2.805842 s	Δ = 102.359 ms Freq = 9.77 Hz					2.908201 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	523.821 µA	223.393 µA	1.235915 mA	2.247551 mA	2.024158 mA	35 nA h	535.998 µA

17 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata37.dlg Trigger Data Log Run Button Source 14585A

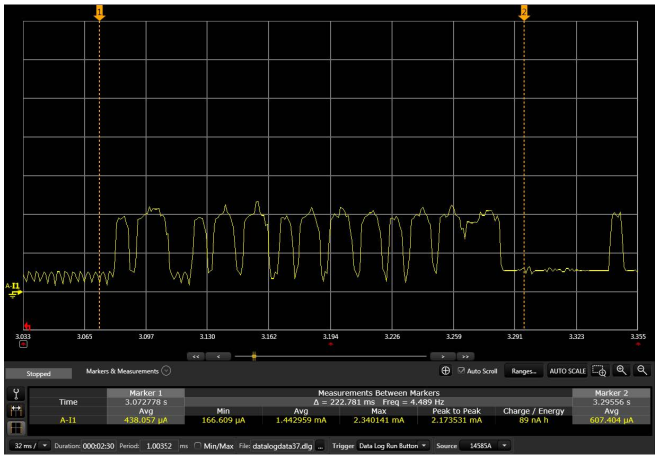
Figure 80. CDLM Initialization State—BMA280 Active Stage Zoom



	Marker 1	Measurements Between Markers					Marker 2
Time	3.072778 s	Δ = 231.813 ms Freq = 4.314 Hz					3.304591 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	223.267 µA	166.609 µA	1.408169 mA	2.340141 mA	2.173531 mA	91 nA h	538.177 µA

68 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata37.dlg Trigger Data Log Run Button Source 14585A

Figure 81. CDLM Initialization State—Timer_A Active Stage



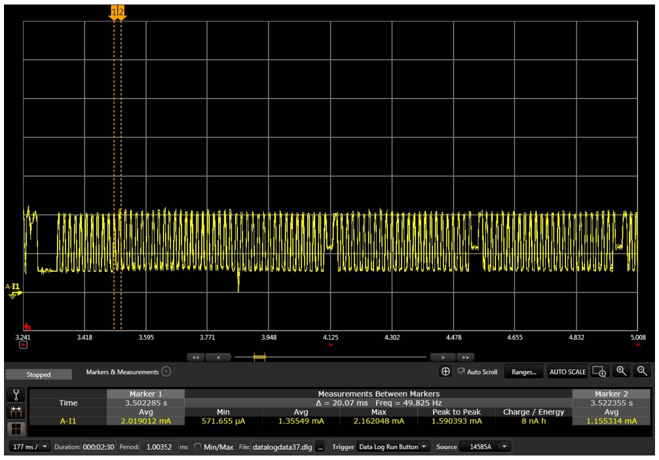
	Marker 1	Measurements Between Markers					Marker 2
Time	3.072778 s	Δ = 222.781 ms Freq = 4.489 Hz					3.29556 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	438.057 µA	166.609 µA	1.442959 mA	2.340141 mA	2.173531 mA	89 nA h	607.404 µA

32 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata37.dlg Trigger Data Log Run Button Source 14585A

Figure 82. CDLM Initialization State—Timer_A Active Stage Zoom

9.5.2 CDLM Execution State

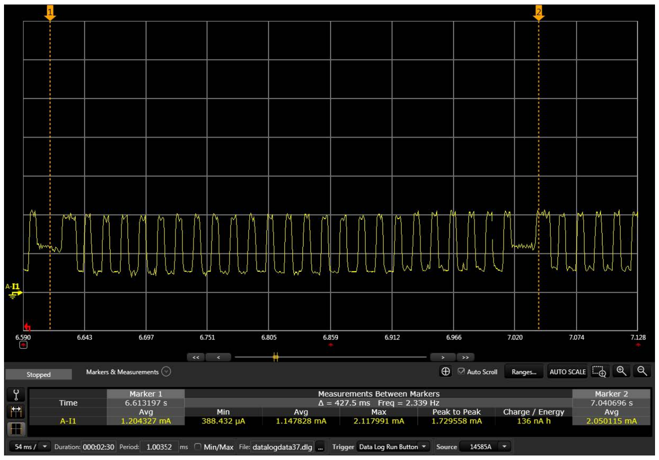
Unfortunately, the BMA280 cannot be disabled when the MPBSM system is set to CDLM mode. The main reason is because the Cadence algorithm requires that the accelerometer be providing an ADC sample 64 times per second. To reduce the possibility of data corruption, and at the cost of power consumption, the BMA280 IC remains in active mode throughout CDLM execution state The time between the temperature readings is set at 1 per second. The Timer_A peripheral is in active mode during the entire duration of CDLM execution state. During CDLM execution state the analytical current consumption is 930 uAmps plus Timer_A and USCI current.



	Marker 1	Measurements Between Markers					Marker 2
Time	3.502285 s	Δ = 20.07 ms Freq = 49.825 Hz					3.522355 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	2.019012 mA	571.655 µA	1.35549 mA	2.162048 mA	1.590393 mA	8 nA h	1.155314 mA

177 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata37.dlg Trigger Data Log Run Button Source 14585A

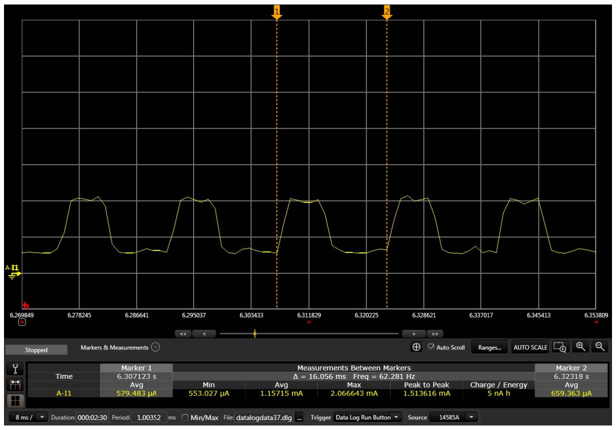
Figure 83. CDLM Execution State—High View



	Marker 1	Measurements Between Markers					Marker 2
Time	6.613197 s	Δ = 427.5 ms Freq = 2.339 Hz					7.040696 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	1.204327 mA	388.432 µA	1.147828 mA	2.117991 mA	1.729558 mA	136 nA h	2.050115 mA

54 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata37.dlg Trigger Data Log Run Button Source 14585A

Figure 84. CDLM Execution State—Medium View



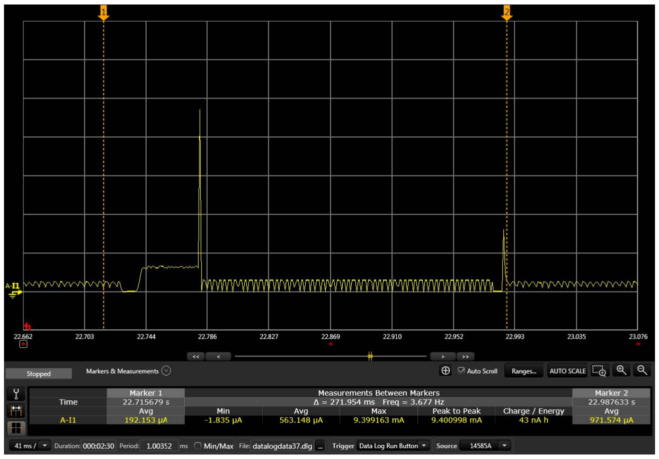
	Marker 1	Measurements Between Markers					Marker 2
Time	6.307123 s	Δ = 16.056 ms Freq = 62.281 Hz					6.32318 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	579.483 µA	553.027 µA	1.15715 mA	2.066643 mA	1.513616 mA	5 nA h	659.363 µA

8 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata37.dlg Trigger Data Log Run Button Source 14585A

Figure 85. CDLM Execution State—Low View

9.5.3 CDLM Power Down State

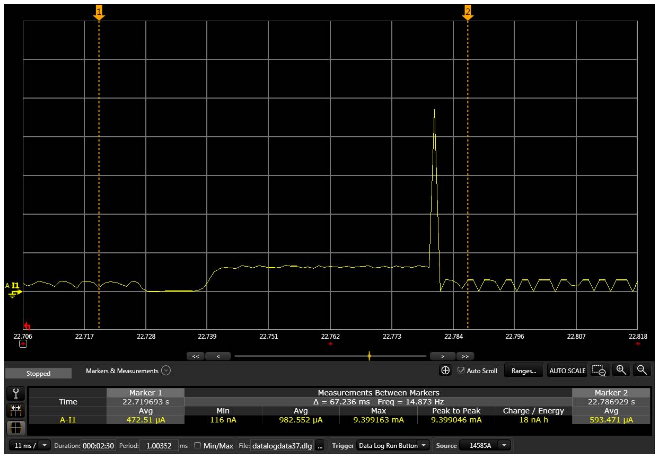
During CDLM power down, both the BMA280 and Timer_A are once again placed in standby mode.



	Marker 1	Measurements Between Markers					Marker 2
Time	22.715679 s	Δ = 271.954 ms Freq = 3.677 Hz					22.987633 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	192.153 µA	-1.835 µA	563.148 µA	9.399163 mA	9.400998 mA	43 nA h	971.574 µA

41 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata37.dlg Trigger Data Log Run Button Source 14585A

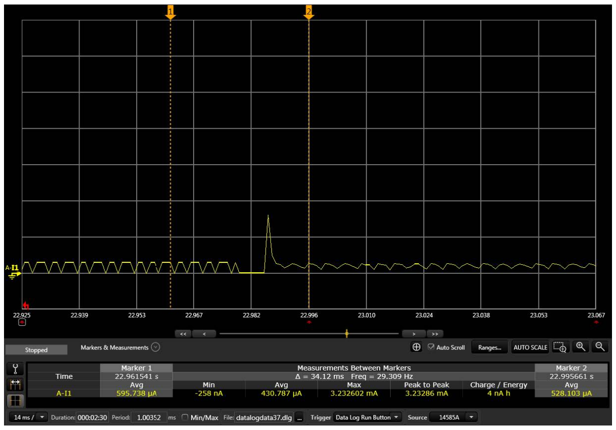
Figure 86. CDLM Power Down State—High View



	Marker 1	Measurements Between Markers					Marker 2
Time	22.719693 s	Δ = 67.236 ms Freq = 14.873 Hz					22.786929 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	472.51 µA	116 nA	982.552 µA	9.399163 mA	9.399046 mA	18 nA h	593.471 µA

11 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata37.dlg Trigger Data Log Run Button Source 14585A

Figure 87. CDLM Power Down State—Disable BMA280 Stage



	Marker 1	Measurements Between Markers					Marker 2
Time	22.961541 s	Δ = 34.12 ms Freq = 29.309 Hz					22.995661 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	Avg
A-I1	595.738 µA	-258 nA	430.787 µA	3.232602 mA	3.23286 mA	4 nA h	528.103 µA

14 ms / Duration: 000:02:30 Period: 1.00352 ms Min/Max File: datalogdata37.dlg Trigger Data Log Run Button Source 14585A

Figure 88. CDLM Power Down State—Disable Timer Stage

10 Design Files

10.1 Schematics

To download the schematics, see the design files at TIDM-BIOSIGNMONITOR.

10.2 Bill of Materials

To download the bill of materials (BOM), see the design files at TIDM-BIOSIGNMONITOR.

10.3 Layer Plots

To download the layer plots, see the design files at TIDM-BIOSIGNMONITOR.

10.4 Altium Project

To download the Altium project files, see the design files at TIDM-BIOSIGNMONITOR.

10.5 Layout Guidelines

To download the layout guidelines, see the design files at TIDM-BIOSIGNMONITOR.

10.6 Gerber Files

To download the Gerber files, see the design files at TIDM-BIOSIGNMONITOR.

10.7 Assembly Drawings

To download the assembly drawings, see the design files at TIDM-BIOSIGNMONITOR.

10.8 Software Files

To download the software files, see the design files at TIDM-BIOSIGNMONITOR.

11 References

1. *RF430FRL152H Firmware User's Guide* (SLAU603)
2. *Galvanic Skin Response*, Wikipedia (http://en.wikipedia.org/wiki/Galvanic_skin_response)
3. *Arousal*, Wikipedia (http://en.wikipedia.org/wiki/Arousal)
4. Design of an NFC Enabled Bio-System Solution by Ricardo Zepeda and Rafael Marquez
5. *How delta-sigma ADCs work, Part 1* by Bonnie Baker
6. *How delta-sigma ADCs work, Part 2* by Bonnie Baker
7. *Decimate*, Merriam-Webster (http://www.merriam-webster.com/dictionary/decimate)
8. Understanding Lead-Off Detection in ECG by Anthony Calabria
9. *Low-Power, 2-Channel, 24-Bit Analog Front-End for Biopotential Measurements* (ADS1292)
10. *LM4041-N/LM4041-N-Q1 Precision Micropower Shunt Voltage Reference* (SNOS641)
11. Single Micropower, 1.6 V, Precision Operational Amplifier with CMOS Inputs (LMP2231)
12. Bosch® Sensortec BMA280 (https://www.bosch-sensortec.com/en/homepage/products_3/3_axis_sensors/acceleration_sensors/bma280/bma280)
13. Spectra® 360 Electrode Gel by Parker Laboratories, Inc (http://www.parkerlabs.com/spectra-360.asp)

11.1 Trademarks

12 About the Author

RICARDO ZEPEDA is an applications engineer at Texas Instruments. Ricky's specialties include but not limited to schematic design, layout design, firmware development, and Android app development. At the time of this writing, he has been working at TI for 4 years. Ricky earned his Bachelor of Science in Electrical Engineering (BSEE) and Master of Science in Computer Engineering from University of Texas at El Paso (UTEP).