TI Designs

NFC Authentication for an EV Charging Station (Pile) Reference Design

Description

The TIDC-EVSE-NFC showcases a straightforward integration of TI’s NFC technology with an existing EVSE platform to enable user authentication. The NFC standard comprises many different sub-technologies and working groups, making integration into a specific system difficult. My leveraging the proven reference designs, and certified software in this design, we can ensure that the standards are met, as well as simplify the development process. This design incorporates only basic functionality for an operational system, but the example software included shows how simple it is to integrate NFC into an EVSE design.

Resources

TIDC-EVSE-NFC	Design Folder
TRF7970A	Product Folder
OPA171	Product Folder
UCC28910	Product Folder
TPS62063	Product Folder
LM7322	Product Folder
TPL7407L	Product Folder
MSP430F6736	Product Folder

TI E2E™ Community

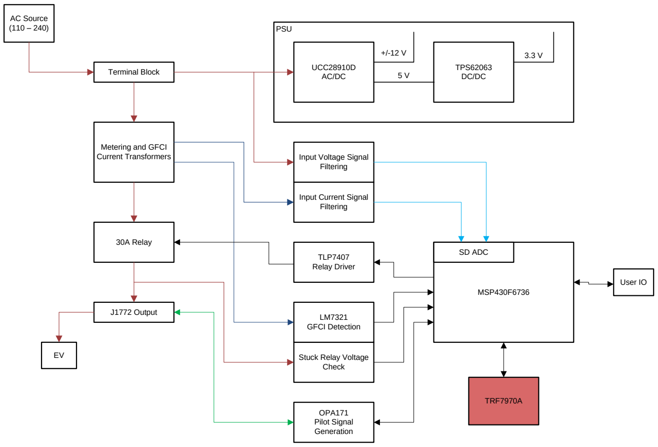
ASK Our E2E Experts

Features

- Ability to Read and Write Near Field Communication (NFC) Type 2, 3, 4A, 4B, and 5 Tag Platforms
- Drop in Integration of NFC Reader to Existing Platform
- Offers a Flexible Firmware Structure That Allows for Configurable NDEF and Custom Proprietary Applications
- Full Implementation of a J1772-Compliant Service Station
- Standardized Pilot Wire Signaling Protocol
- Integrated Utility Meter Grade Energy Measurement

Applications

- Level 1 and Level 2 Electric Vehicle Service Equipment (EVSE)



Copyright © 2016, Texas Instruments Incorporated

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

1 System Overview

1.1 System Description

This design implements a basic EVSE with NFC card reading functionality integrated on the same platform. The NFC reader demonstrates a simple authentication mechanism for public access EVSEs or has the capability to also add a read or write function for local prepayment debiting. Host public EVSEs, however, use a simple read mechanism, as shown in this design, and authenticate the NFC card data against an external could-based service.

Electric vehicles (EVs) have been around for over a century but have had limited market success until recently. Modern advances in battery technology drive efficiency and market forces have accelerated the demand and deployment rates of EVs that can compete with traditional internal combustion vehicles. Many vehicle owners now have a requirement of plugging and unplugging a very high-power device to their home electricity system on a regular basis.

Traditional international combustion vehicles also benefit from a broad network of gas stations for rapid energy delivery to the vehicle and range extension. While the technology is improving, EVs still suffer from slow energy delivery rates, which require vehicles to be stationary for long periods of time to recharge. The slow energy delivery systems behind EVs also highlight the shortcomings of the gas station model.

The slow rate of charging is the result of using low battery charge currents to avoid damage (a problem that is constantly being improved), as well as the energy capacity of the local grid connection. Adding a high power connection can create issues in regards to safety and reliability. Public charge stations are able to tap into much higher current connections available in commercial buildings; however, these stations must be able to charge all varieties of EVs on the road to be resourceful.

Such problems are being mitigated through EVSE, which controls the power flow into an EV. Many vehicle manufacturers have adopted the J1772 SAE standard for AC electrical connections to a vehicle. The same specifications also translate into international localizations with only differing form factors.

The standard design of EV charging systems on the market at the time of this writing have the AC-DC converter for the battery charge system integrated into the vehicle, so only AC power is required. External DC-DC and charge circuitry is enabled on some vehicles, but this configuration is outside the scope of this design. To facilitate the power delivery to the vehicle, the EVSE sits between a stable grid connection and the vehicle, as shows.

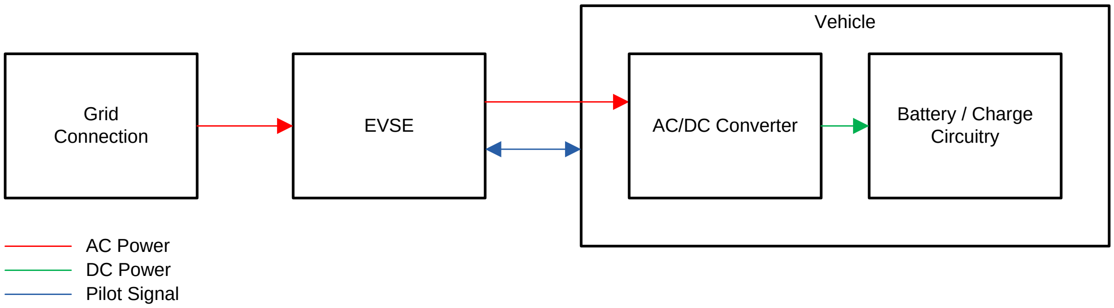


Figure 1. EVSE Position in Power Flow

The core of the EVSE operation for AC power delivery is communication with the vehicle over a single-line pilot wire. This EVSE operation is based on level 1 and level 2 devices. Table 1 describes the differences in level 1 and level 2 devices. This single-line pilot wire enables negotiation with the vehicle for power status, available power, and charge state. In addition, the EVSE must be able to control AC power delivery to the plug itself (up to 240-V AC and 80 A in some cases) to necessitate robust relay or contactor driving.

Table 1. Level 1 and Level 2 Charging Standards

LEVEL	DEFINITION	ENERGY RATING
Level 1	Alternating current energy to the onboard charger of the vehicle; from the most common U.S. grounded household receptacle, commonly referred to as a 120-V outlet.	120-V AC; 16 A (= 1.92 kW)
Level 2	Alternating current energy to the onboard charger of the vehicle; 208 V to 240-V, single phase. The maximum current specified is 32 A (continuous) with a branch circuit breaker rated at 40 A.	208- to 240-V AC; 12 A to 80 A (= 2.5 kW to 19.2 kW)

This implementation of an EVSE contains a basic set of features, which are expandable to enable additional usage scenarios. The primary functionality includes:

- Level 1 and level 2 operation (120 to 240 V)
- Power delivery up to 30 A (expandable with larger relays)
- Pilot signal wire communication support
- Latched relay detection
- Energy metering

A normal residential instillation will be physically access controlled and will not typically require a means of local user authentication. However, in public spaces, many EVSE owners will want some mechanism of controlling station use. There are many reasons behind this, but the primary reason is for billing to recoup instillation or energy usage costs. The most common form of authentication for this style of station is NFC, where a user simply taps a registered card, and the station will authenticate the user against a cloud based server through GSM or GPRS.

2 Block Diagram

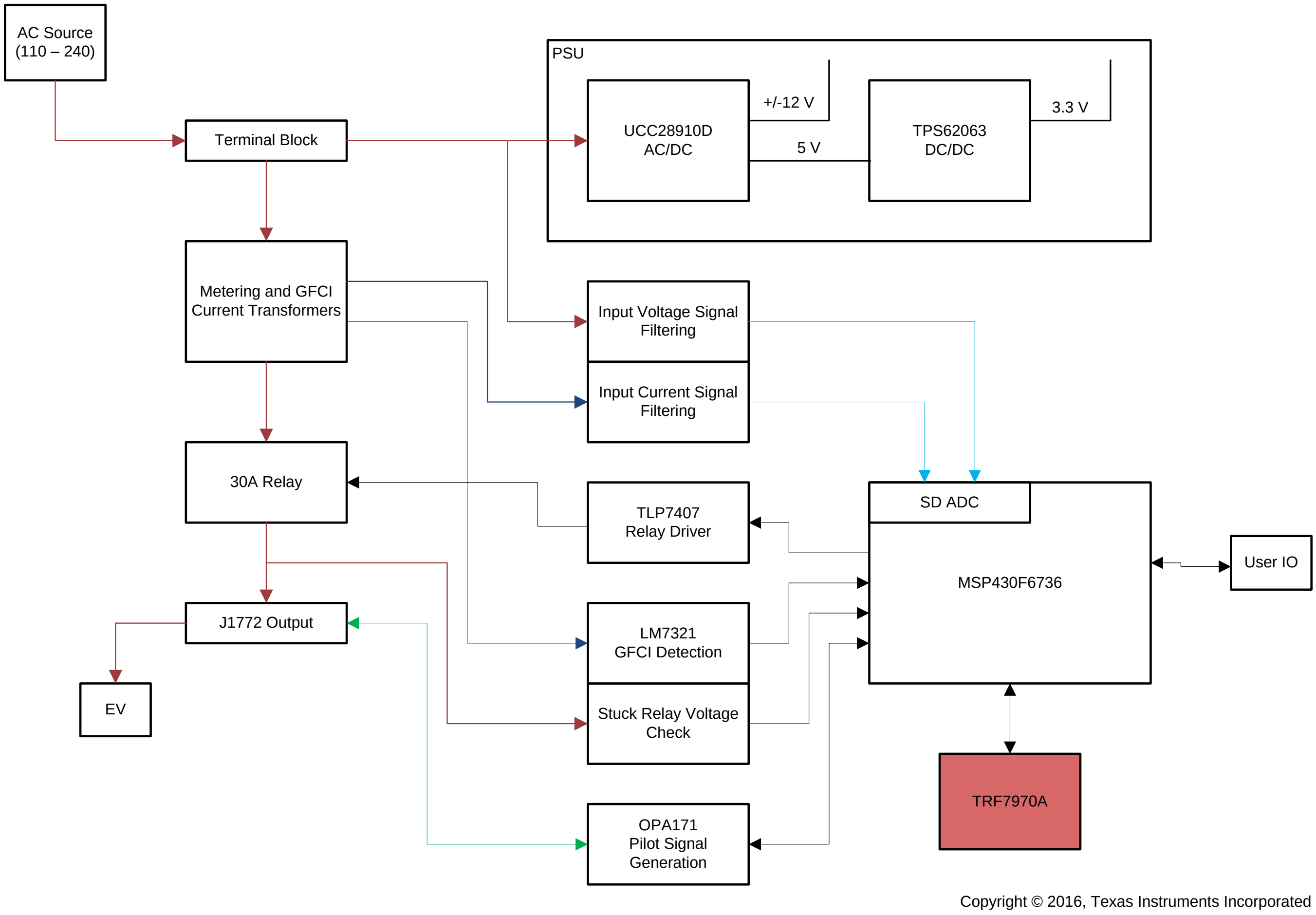


Figure 2. Block Diagram

2.1 *Highlighted Products*

The TIDC-EVSE-NFC reference design features the following devices:

- **TRF7970A**: multi-protocol fully integrated 13.56-MHz NFC and RFID transceiver
- **MSP430F6736**: mixed-signal microcontroller (MCU)
- **UCC28910D**: 700-V flyback switcher with a constant-voltage (CV), constant-current (CC) and primary-side control
- **TPS62063**: 3-MHz, 1.6-A step-down converter
- **OPA171**: 36-V, low-power, rail-to-rail output (RRO), general purpose operational amplifier (op amp)
- **LM7322**: high output current and unlimited cap load ±15-V op amp
- **TPL7407**: 40-V, 7-channel NMOS array, low-side driver

For more information on each of these devices, see the respective product folders at www.ti.com.

2.1.1 TRF7970A Features

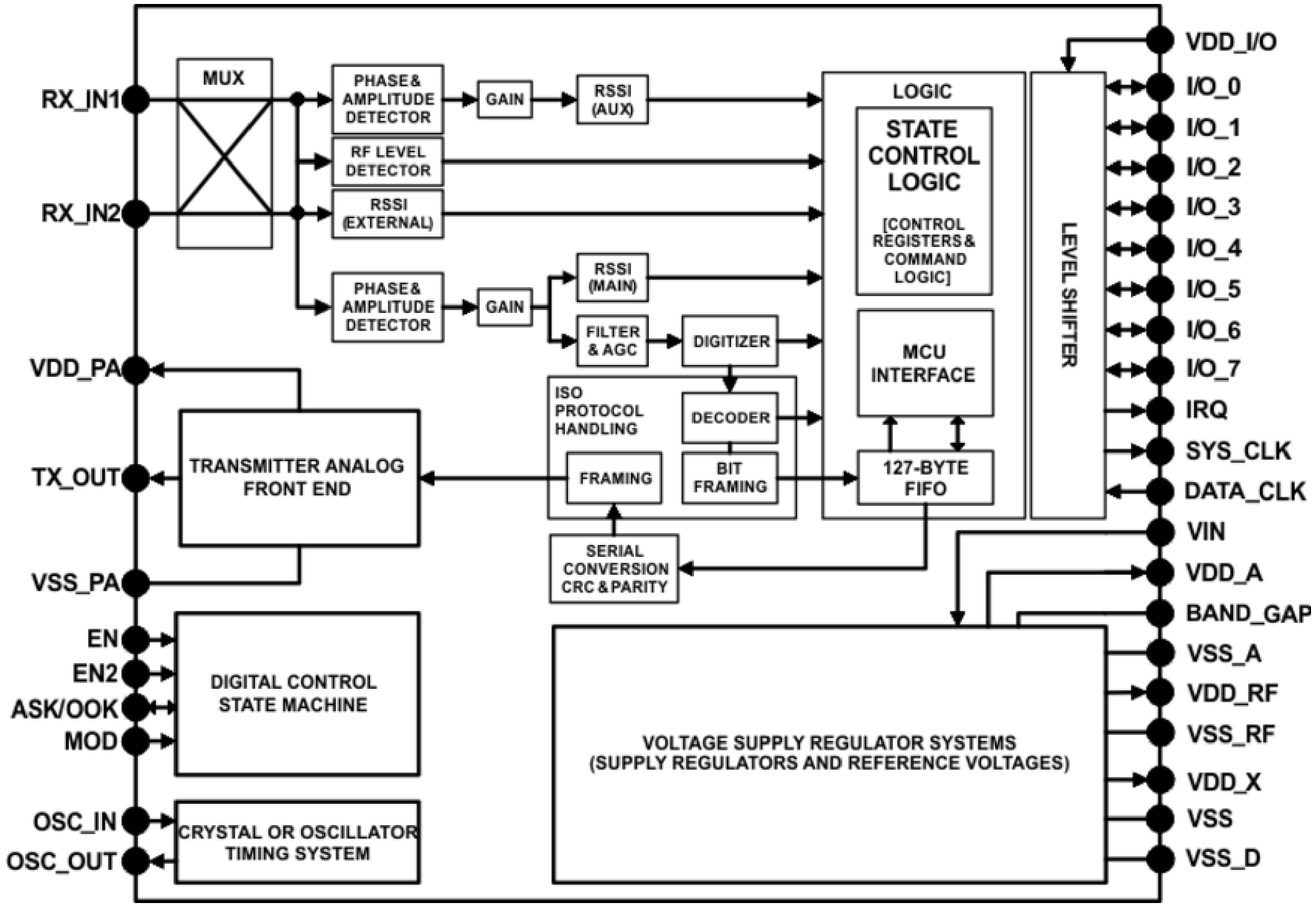


Figure 3. TRF7970A Hardware Overview

- Supports NFC Standards NFCIP-1 (ISO/IEC 18092) and NFCIP-2 (ISO/IEC 21481)
- Completely Integrated protocol handling for ISO15693, ISO18000-3, ISO14443A/B, and FeliCa
- Integrated encoders, decoders, and data framing for NFC initiator, active and passive target operation for all three bit rates (106 kbps, 212 kbps, 424 kbps) and card emulation
- RF field detector with programmable wake-up levels for NFC passive transponder emulation operation
- RF field detector for NFC physical collision avoidance
- Integrated state machine for ISO14443A anticollision (broken bytes) operation (transponder emulation or NFC passive target)
- Input voltage range: 2.7 to 5.5 VDC
- Programmable output power: +20 dBm (100 mW), +23 dBm (200 mW)
- Programmable I/O voltage levels from 1.8 to 5.5 VDC
- Programmable system clock frequency output (RF, RF/2, RF/4) from 13.56-MHz or 27.12-MHz crystal or oscillator
- Integrated voltage regulator output for other system components (MCU, peripherals, indicators), 20 mA (maximum)
- Programmable modulation depth
- Dual receiver architecture with RSSI for elimination of "read holes" and adjacent reader system or ambient in-band noise detection
- Programmable power modes for ultra low-power system design (power down <1 µA)

- Parallel interface or serial peripheral interface (SPI) (with 127-Byte FIFO)
- Temperature range: –40°C to 110°C
- 32-pin QFN package (5 mm × 5 mm)

2.1.2 MSP430F6736 Features

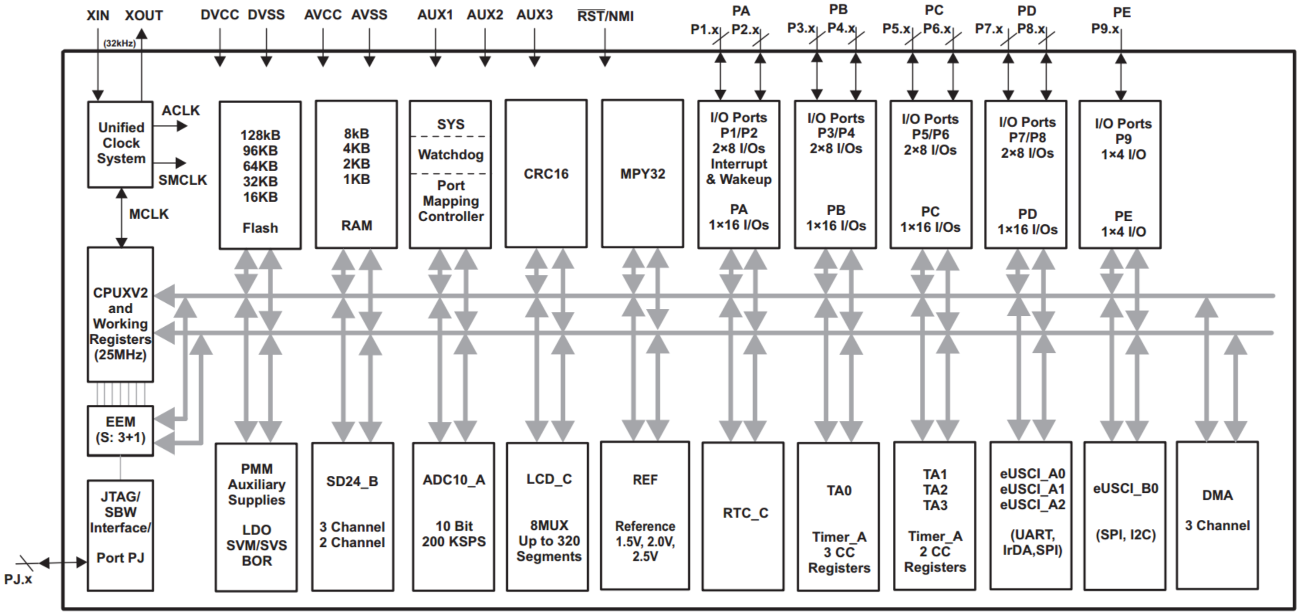

XIN
XOUT
(32kHz)
DVCC DVSS
AVCC AVSS
AUX1 AUX2 AUX3
RST/NMI
PA P1.x P2.x
PB P3.x P4.x
PC P5.x P6.x
PD P7.x P8.x
PE P9.x
Unified Clock System
ACLK
SMCLK
MCLK
128kB 96KB 64KB 32KB 16KB Flash
8kB 4KB 2KB 1KB RAM
SYS Watchdog Port Mapping Controller
CRC16
MPY32
I/O Ports P1/P2 2×8 I/Os Interrupt & Wakeup PA 1×16 I/Os
I/O Ports P3/P4 2×8 I/Os PB 1×16 I/Os
I/O Ports P5/P6 2×8 I/Os PC 1×16 I/Os
I/O Ports P7/P8 2×8 I/Os PD 1×16 I/Os
I/O Ports P9 1×4 I/O PE 1×4 I/O
CPUXV2 and Working Registers (25MHz)
EEM (S: 3+1)
JTAG/ SBW Interface/ Port PJ
PJ.x
PMM Auxiliary Supplies LDO SVM/SVS BOR
SD24_B 3 Channel 2 Channel
ADC10_A 10 Bit 200 KSPS
LCD_C 8MUX Up to 320 Segments
REF Reference 1.5V, 2.0V, 2.5V
RTC_C
TA0 Timer_A 3 CC Registers
TA1 TA2 TA3 Timer_A 2 CC Registers
eUSCI_A0 eUSCI_A1 eUSCI_A2 (UART, IrDA,SPI)
eUSCI_B0 (SPI, I2C)
DMA 3 Channel

Figure 4. MSP430F6736 Block Diagram

- Low supply voltage range: 3.6 V down to 1.8 V
- Ultra-low power consumption
- Active mode (AM): all system clocks active 265 µA/MHz at 8 MHz, 3.0 V, flash program execution (typical) 140 µA/MHz at 8 MHz, 3.0 V, RAM program execution (typical)
- Standby mode (LPM3): real-time clock (RTC) with crystal, watchdog, and supply supervisor operational, full RAM retention, fast wake-up: 1.7 µA at 2.2 V, 2.5 µA at 3.0 V (typical)
- Off mode (LPM4): full RAM retention, supply supervisor operational, fast wake-up: 1.6 µA at 3.0 V (typical)
- Shutdown RTC mode (LPM3.5): shutdown mode, active RTC with crystal: 1.24 µA at 3.0 V (typical)
- Shutdown mode (LPM4.5): 0.78 µA at 3.0 V (typical)
- Wake-up from standby mode in 3 µs (typical)
- 16-bit RISC architecture, extended memory, up to 25-MHz system clock
- Flexible power management system
- Fully integrated LDO with programmable regulated core supply voltage
- Supply voltage supervision, monitoring, and brownout
- System operation from up to two auxiliary power supplies
- Unified clock system
- Frequency-locked loop (FLL) control loop for frequency stabilization
- Low-power low-frequency internal clock source (VLO)
- Low-frequency trimmed internal reference source (REFO)
- 32-kHz crystals (XT1)
- One 16-bit timer with three capture and compare registers

- Three 16-bit timers with two capture and compare registers each
- Enhanced universal serial communication interfaces
- eUSCI_A0, eUSCI_A1, and eUSCI_A2: enhanced universal asynchronous receiver/transmitter (UART) supports auto-baudrate detection, IrDA encoder and decoder, synchronous SPI
- eUSCI_B0: I^{2}C with multi-slave addressing, synchronous SPI
- Password-protected RTC with crystal offset calibration and temperature compensation
- Separate voltage supply for backup subsystem
- 32-kHz low-frequency oscillator (XT1)
- RTC
- Backup memory (4 bits × 16 bits)
- Three 24-bit ∆∑ analog-to-digital converters (ADCs) with differential PGA inputs
- Integrated liquid-crystal display (LCD) driver with contrast control for up to 320 segments in 8-mux mode
- Hardware multiplier supports 32-bit operations
- 10-bit 200-kSPS A-D converter
- Internal reference
- Sample-and-hold, auto-scan feature
- Up to six external channels, two internal channels, Including temperature sensor
- Three-channel internal DMA
- Serial onboard programming, no external programming voltage required
- Available in 100-pin and 80-pin LQFP packages

2.1.3 UCC28910D Features

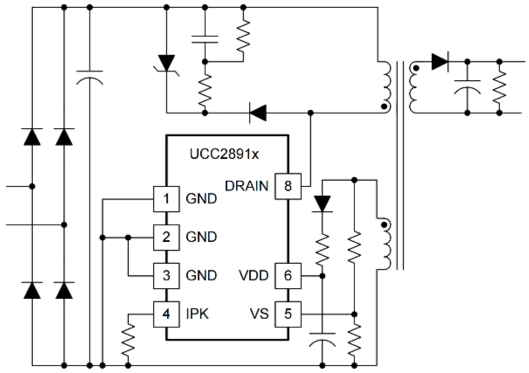


Figure 5. UCC28910D Simplified Schematic

- CV and CC output regulation without optical-coupler
- ±5% output voltage regulation accuracy
- ±5% output current regulation with AC line and primary inductance tolerance compensation
- 700-V start-up and smart power management enables <30-mW standby power
- 115-kHz maximum switching frequency design for high-power density
- Valley switching and frequency dithering to ease electromagnetic interference (EMI) compliance
- Thermal shut down
- Low line and output over-voltage protection

2.1.4 TPS62063 Features

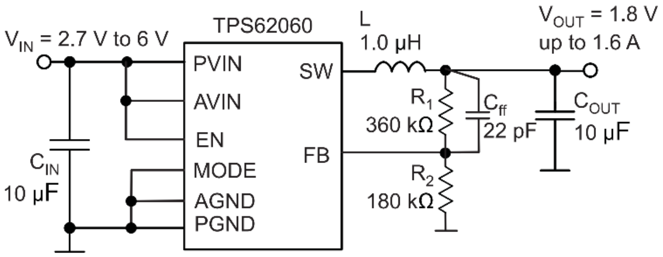


Figure 6. TPS62063 Typical Schematic

- 3-MHz switching frequency
- VIN range from 2.7 to 6 V
- 1.6-A output current
- Up to 97% efficiency
- Power save mode and 3-MHz fixed pulse width modulation (PWM) mode
- Output voltage accuracy in PWM mode ±1.5%
- Output discharge function
- Typical 18-µA quiescent current
- 100% duty cycle for lowest dropout
- Voltage positioning
- Clock dithering
- Supports maximum 1-mm height solutions
- Available in a 2 mm × 2 mm × 0.75 mm WSON

2.1.5 OPA171 Functionality

- Supply range: 2.7 V to 36 V, ±1.35 V to ±18 V
- Low noise: 14 nV/√Hz
- Low offset drift: ±0.3 µV/°C (typical)
- RFI filtered inputs
- Input range includes the negative supply
- Input range operates to positive supply
- RRO
- Gain bandwidth: 3 MHz
- Low quiescent current: 475 µA per amplifier
- High common-mode rejection: 120 dB (typical)
- Low-input bias current: 8 pA

2.1.6 LM7322 Features

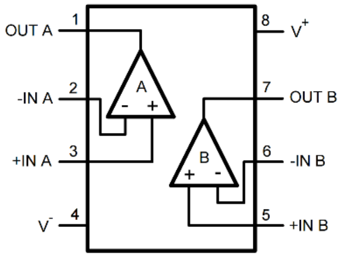


Figure 7. LM7322 Package Layout

- VS = ±15, TA = 25°C (typical values unless specified)
- Wide supply voltage range: 2.5 to 32 V
- Output current: +65 mA or –100 mA
- Gain bandwidth product: 20 MHz
- Slew rate: 18 V/µs
- Capacitive load tolerance unlimited
- Input common-mode voltage 0.3 V beyond rails
- Input voltage noise: 15 nV/√Hz
- Input current noise: 1.3 pA/√Hz
- Supply current and channel 1.1 mA
- Distortion THD+noise: –86 dB
- Temperature range: –40°C to 125°C
- Tested at –40°C, 25°C, and 125°C at 2.7 V, ±5 V, and ±15 V
- LM732xx are automotive grade products that are AEC-Q100 grade 1 qualified

2.1.7 TPL7407 Features

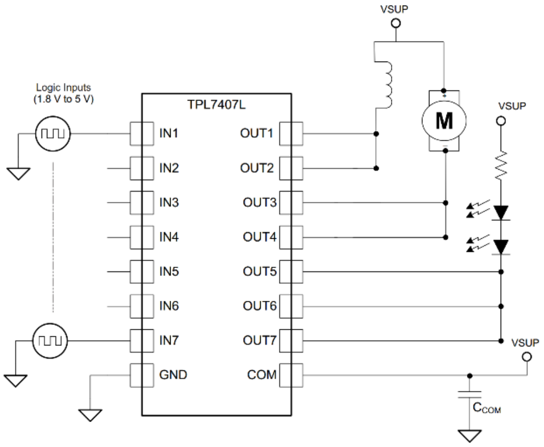


Figure 8. TPL7407 Sample Application Schematic

- 600-mA rated drain current (per channel)
- CMOS pin-to-pin improvement of seven-channel Darlington array (for example, ULN2003A)
- Power efficient (very low VOL)
- Very low output leakage <10-nA per channel

- Extended ambient temperature range: TA = –40°C to 125°C
- High-voltage outputs: 40 V
- Compatible with 1.8- to 5.0-V MCUs and logic interface
- Internal free-wheeling diodes for inductive kick-back protection
- Input pulldown resistors allows tri-stating the input driver
- Input RC-snubber to eliminate spurious operation in noisy environments
- Inductive load driver applications
- Electrostatic discharge (ESD) protection exceeds JESD 22

3 System Design Theory

3.1 Pilot Signal Interface

The pilot signal is the key method in which a J1772-compliant EVSE communicates with a vehicle. The pilot signal is based on a 1-kHz, ±12-V PWM signal that is transmitted to a vehicle over the charge chord. The vehicle can then respond by placing various loads on the line, which affects the vehicle's voltage that the EVSE measures.

3.1.1 J1772 Duty Cycle

The duty cycle of the pilot signals communicates the limit of current the EVSE is capable of supplying to the vehicle; the vehicle can then use up to that amount of current for its charging circuitry. This current rating is primarily determined by the electromechanical components in the EVSE, such as, conductors, relays, contactors, and the service connection.

The relationship between duty cycle and current is defined by two different equations depending on the current range specified; for a 6- to 51-A service, Equation 1 is:

$$\text{DutyCycle} = \frac{\text{Amps}}{0.6} \tag{1}$$

For a higher service in the 51- to 80-A range, Equation 2 is:

$$\text{DutyCycle} = \left(\frac{\text{Amps}}{2.5}\right) + 64 \tag{2}$$

To demonstrate this relationship further, shows some of the common service ratings.

Table 2. Pilot Wire Example Duty Cycles

AMPS	DUTY CYCLE
5	8.3%
15	25%
30	50%
40	66.6%
65	90%
80	96%

In this design, the PWM is generated by a timer module on the MSP430™ MCU. Because the current rating is so tightly coupled to the external hardware, the PWM value can typically be set as a permanent value in the firmware.

Advanced EVSEs with a human machine interface (HMI) can enable the current to be derated if the service line is unable to provide enough current with a stable voltage. A significant voltage drop as a result of wire loss is possible in these high-current applications.

3.1.2 Pilot Signal States

The EVSE connection and negotiation occurs through various states of the PWM signal and load resistances of the vehicle. Table 3 highlights these states:

Table 3. Pilot Wire Example Duty Cycles

STATE	PILOT HIGH VOLTAGE	PILOT LOW VOLTAGE	FREQUENCY	RESISTANCE	DESCRIPTION
State A	12 V	-	DC	-	Not Connected
State B	9 V	-12 V	1 kHz	2.74 kΩ	EV Connected, ready to charge
State C	6 V	-12 V	1 kHz	882 Ω	EV Charging
State D	3 V	-12 V	1 kHz	246 Ω	EV Charging, ventilation required
State E	0 V	0 V	-	-	Error
State F	-	-12 V	-	-	Unknown Error

States A, B, and C are the core functionality and define the normal operation. An EVSE typically performs several self-tests upon initially powering on and then enters State A. When ready, the normal connection process follows several steps.

1. The EVSE puts 12 V on the pilot wire. This transmission signals the vehicle when the plug has been connected.
2. When the plug has been connected, the vehicle places a 2.74-kΩ load on the pilot line, which drops the voltage to 9 V.
3. The EVSE moves to State B, where it enables the PWM, which signals the vehicle how much current it can draw. The EVSE also closes the relays providing power to the vehicle.
4. The vehicle starts to draw power and switches to the 822-Ω load, which drops the voltage to 6 V signaling the EVSE that charging has started.
5. Most vehicles continue to pull low amounts of power in state C, even when fully charged, so the charging process is ended by unplugging the cable, which returns the voltage to 12 V. The EVSE measures this process and closes the relays and returns to State A.

Additional error handling such as missing diodes in the vehicle or an improper connection can be detected and handled by the EVSE by cutting the power, as well.

3.1.3 Pilot Signal Circuit

The pilot signal is required to travel down several meters of cable and through a load resistance. The pilot signal is also a bipolar ±12-V signal, which requires special consideration.

To accommodate these parameters, an amplifier with a wide input range and reasonable power output has been selected. The OPA171 has a voltage rating of ±18 V and a current rating of 475 mA, making it suitable for the application. In addition, while most EVSEs do not require an automotive qualification, a Q1-rated variant of the OPA171 exists if this feature is desired.

The amplification circuit is a simple RRO configuration of the OPA171 device, with the MCU I/O driving the positive input. The output of the pilot amplifier is also fed into a simple voltage divider so that the MCU can measure the voltage during operation and detect the load resistance of the vehicle. shows the full schematic of this subsystem.

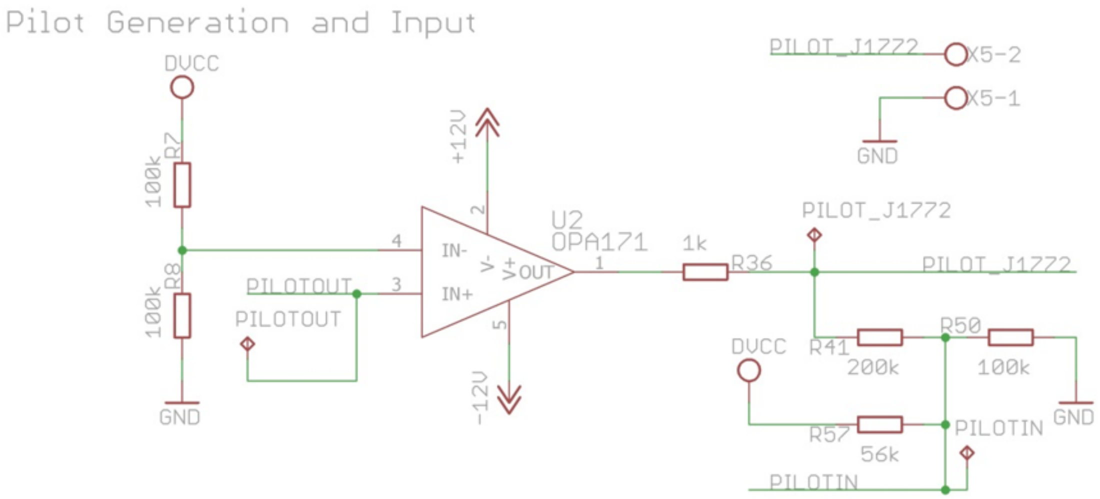


Figure 9. Pilot Signal Generation and Input

To validate the architecture, the design has been tested in the Texas Instrument's TINA-TI™ software, which is a spice-based simulation tool. The resistor load states of the EVSE have also been included in these tests to simulate the response to the state changes. The simulation files are included in the design download packages. shows an example result of the simulation for a State B condition.

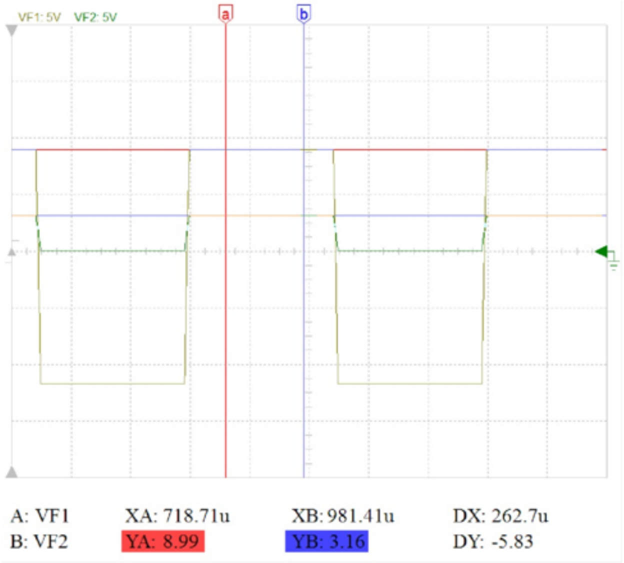


Figure 10. Pilot Signal TINA Simulation

This capture has a 5-V per division Y-axis. VF1 is the line out channel and VF2 is the MCU ADC channel. The user can observe here that marker A (on VF1) is measuring 8.99 V and marker B (on VF2) is measuring 3.16 V. In addition, VF1 shows that the –12-V side is still intact and that VF2 does not drop below 0 V, which could damage the MCU. The pilot output is correct for the application and the MCU can easily measure the incoming signal.

3.2 Relay Drive and Latch Detect

The primary functionality of the EVSE is the reliable control of large currents directed toward an EV at the mains voltage. In a normal use case, the relay must be held closed for several hours to fully charge a vehicle; however the relays cannot be latching because of safety concerns. If something fails in the control system, the relays must fail open. These high current relays can typically draw tens to hundreds of milliamps as an inductive load, which requires specific drive architectures.

Because of the amount of time that a relay requires to remain powered, an efficient drive solution is preferred to the typical Darlington array or even discrete transistor configuration. For this reason, the TPL7404L low-side driver has been selected to drive the relays in the design. The TPL7404L offers high-efficiency, integrated diode protection for inductive loads and has a wide-voltage output capability to match most electromechanical relays. The design defaults to a 5-V output but an external voltage can be used depending on the relay configuration.

The relay configuration used for design testing contains a two-stage approach. The first relay is controlled by the EVSE board through a 12-V signal from the TPL7404L device. This relay switches a 120-V signal into a much larger relay that is capable of supporting the large currents required of the EVSE. Many large contactors or high-amp relays are 120 V, so this configuration is not uncommon. This configuration also reduces power supply requirements because a lower current is required to drive a smaller first relay. shows a simple diagram of the relay configuration.

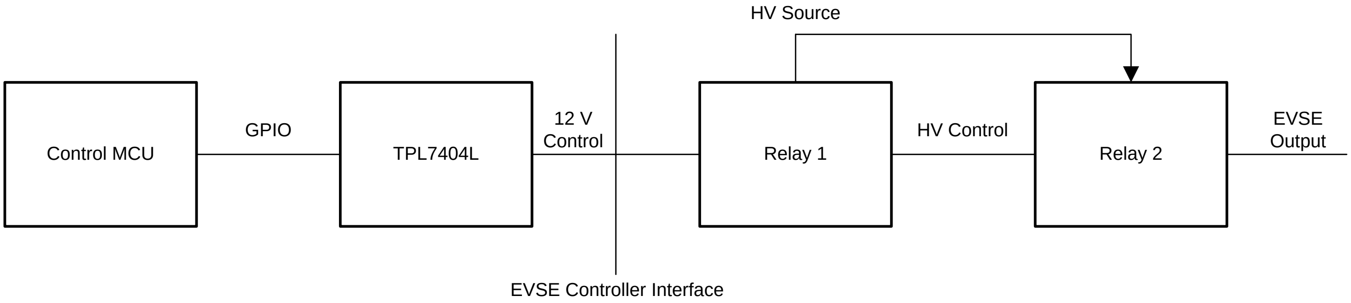


Figure 11. Relay Configuration

For safety reasons, detecting the output voltage of the primary relay is critical. On high voltage relays, the contacts can experience arcing and become fused together, which provides power to the plug even when the system is not powering it. Checking that the operation completed correctly is important and should be done every time the relay is opened. To implement this check, an ADC can be used with a voltage divider, which is the same process used in the energy metering; however, the user can construct a simpler solution from an AC input optocoupler. The output from this is a GPIO-level DC signal that is high when voltage is present and can be fed directly into the MCU.

3.3 Energy Metering

The energy measurement section of this hardware design has been emulated from TI’s existing portfolio of residential e-meter designs, specifically, the class 0.2 single-phase e-meter (TIDM-SINGLEPHASEMETER-AFE). This design offers a high-accuracy energy measurement through the MSP430F6736 MCU. The MCU is fully programmable and has also been adopted to run the software controlling the EVSE system. For more information about the metering solution, refer to the associated reference design (TIDM-SINGLEPHASEMETER).

4 Getting Started Hardware

4.1 Hardware Overview

The control system for the TIDC-EVSE-NFC EVSE reference design is fully implemented in a single PCB. All schematic, layer prints, bill of materials (BOM), and other design resources are available in the reference design folder. shows the top view of the board.

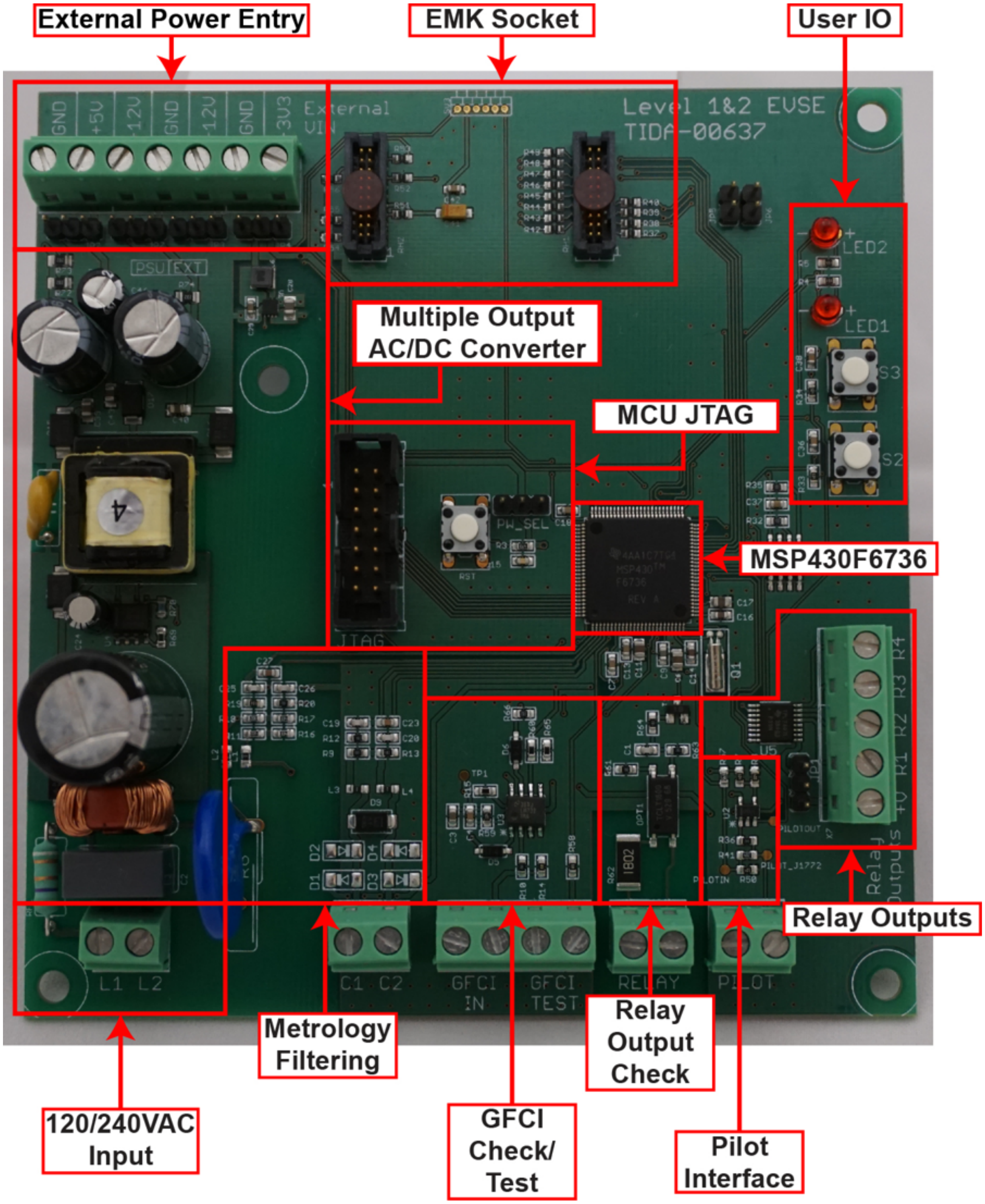


Figure 12. Top View of EVSE PCB

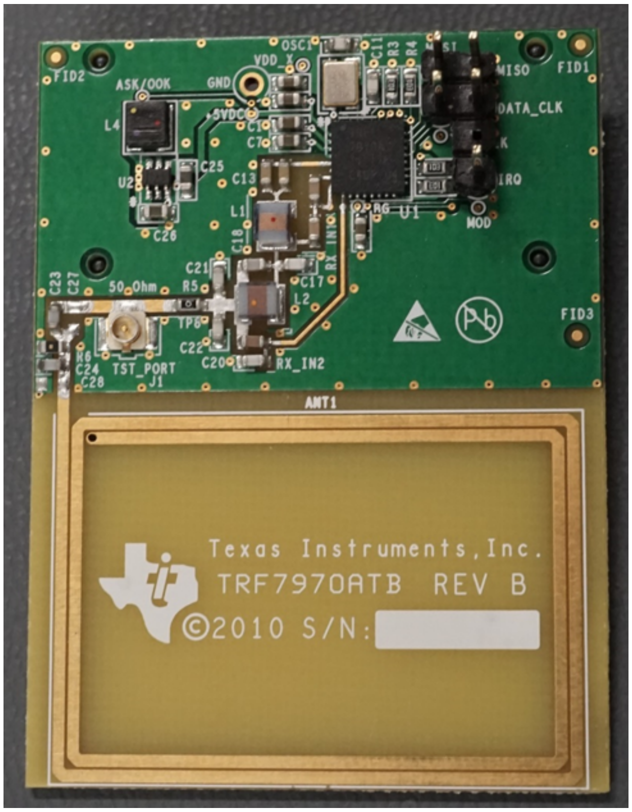


Figure 13. Front View of the TRF7970ATB

The EVSE has the following functional blocks:

1. **External power entry**: Several screw terminals are available to evaluate alternative power options with the PCB. The system requires at least ±12 V and 3.3 V to operate, with an optional 5-V input available. Each input has a jumper option to enable selection between the internal and external power.
2. **EMK socket**: A standard TI EMK socket has been provided to enable the easy addition of wireless cards produced by TI. In addition, is the board provides a second, dedicated communication power for EZ-RF boards or other UART compatible devices.
3. **User I/O**: Two onboard buttons and two light-emitting diodes (LEDs) have been provided to provide a simple user interface with the system. By default, the buttons do not have mapped functions, and the LEDs signal error states for the GFCI and pilot line systems.
4. **MSP430F6736**: The F6736 series MSP430 is the MCU used to control the EVSE.
5. **MCU JTAG**: A standard MSP430 14-pin JTAG connector is available for programing the device. A power selection jumper (to enable the MCU to be powered from the MSPFET) and reset push button have also been included.
6. **Multiple output AC-DC converter**: The onboard AC-DC converter is capable of producing ±12 V and 5 V from a 120- or 240-V AC input. A small footprint DC-DC converter produces the 3.3-V rail from the 5-V output. All of these rails can be bypassed and powered externally through the external power entry block.
7. **120- to 240-V AC input**: The AC power entry terminal supports a wide input voltage and feeds both the onboard power supply as well as the metrology section of the system.
8. **Metrology filtering**: The signal filtering for the energy metering is based on TI's class 0.2 single-phase e-meter (TIDM-SINGLEPHASEEMETER-AFE). The design is a simple passive filter for both the AC voltage and a current transformer input through the terminal blocks. An external 0.05% CT is required to meet the high accuracy specification with the appropriate burden resistor tuning. Section 5.2 provides additional information on the functionality of this hardware.

9. **GFCI check and GFCI test**: Filtering for the GFCI input signal and the test signal output is supplied here. Section 5.3 provides additional information on the hardware connections and setup.
10. **Relay output check**: The AC output of the relay can be fed back into the EVSE for monitoring. This enables a quick check of the relay functionality.
11. **Pilot interface**: A screw terminal is available here to make a connection between the pilot signal to the amplification and filtering block. provides additional information on the functionality of this hardware.
12. **Relay outputs**: Output capability is provided for up to four discrete relay signals. An onboard jumper is available to select between using the onboard power supply to drive the relay and TPL7407 or using an external voltage.

The NFC functionality is added through the TRF7970ATB daughter card with fully integrated smart NFC transceiver system onboard. This card is shown in . Additional technical resources pertaining to this design can be found in the *TRF79x0ATB NFC/HF RFID Reader Module User's Guide*.

4.2 Mains Power and Metrology

The power entry block on the TIDA-00637 reference design serves to both power the system and provide a signal to the metrology filters to be measured by the MCU. On a single-phase feeder, the terminal assignment is: Live → L1 and Neutral → L2. On a split-phase connection, the terminal connection does not have a specific polarity. In both cases, a ground connection is required but only for the pilot wire signal, as Section 5.4 details.

To measure the current that the vehicle is drawing, the terminal block labeled C1 and C2 is used to input the current output of a current transformer. The system has an integrated 13-Ω burden resistor and a 2000:1 ratio CT is typically used to give accurate metrology measurements up to 100 A with a 2000:1 dynamic range. If a lower ceiling on the current range is required, a CT with a lower current rating can be used, but the burden resistor must be chosen to match and the software must be updated with the new ranging information. The class 0.2 single-phase e-meter design provides additional information on choosing a CT (TIDM-SINGLEPHASEEMETER-AFE).

4.3 Pilot Wire

The fourth and smaller wire on a standard J1772 cable is the pilot line. This line can be electrically connected directly to the right side of the *PILOT* terminal block on the EVSE system design. The EV interacts with the pilot signal by placing a resistance between the pilot pin and the ground connection on the cable. While the ground connection is not used for the power elements in this design, it is required as an electrical reference for the pilot signal. To set the ground connection to act as the electrical reference, the left connection of the *PILOT* terminal block must be connected to the earth ground on the EVSE cable.

4.4 Relays

The relay terminal block on the EVSE system design has output support for four relays, with a fifth terminal for a positive voltage output from the onboard supply, or external supply input. The four relay connectors have been set up to enable low-side switching of the relays without the requirement for external snubber diodes; however, having the diodes in place does not affect operation.

For normal connection of a relay using the onboard 12-V rail, the relay coil is connected between the +V port, and the associated relay port on the terminal block, as shows.

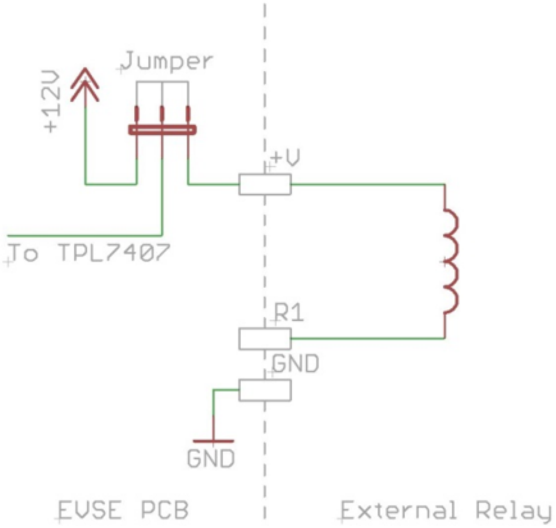


Figure 14. Normal Relay Connection

To use an external supply, the positive rail must connect to the +V port, ground to the GND port, and the relay to the associated control port, as shows.

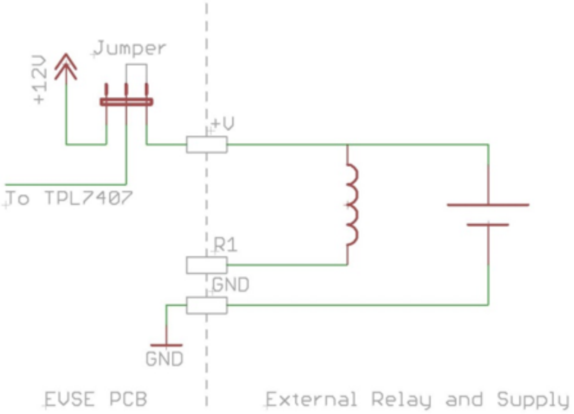


Figure 15. Relay with External Supply

In many cases, a relay able to switch the amount of current required for an EVSE requires a significant level of switching current to latch. Many high current relays or contacts mitigate this requirement by using line-level AC voltage on the coil. To control this style of relay or contactor, a smaller relay with a low-voltage control level can be switched by the EVSE, which, in turn, switches the line voltage to the high current control.

4.5 TRF7970ATB Daughter Card Interface

To add NFC capabilities to the TIDC-EVSE-NFC, the TRF7070ATB evaluation module is connected to the EVM. The TRF7970ATB contains a TRF7970A NFC transceiver, as well as all supporting electronics, and a built in antenna and matching network. The daughter card is interfaced with via the TI standard EMK headers and pinout. To connect the TRF7970ATB to the EVM, the EM female connector is placed in the TIDC-EVSE-NFC's RF connector. Care should be taken to not damage the small pitched pins in the headers.

Figure 16 shows the connections made from the TIDC-EVSE-NFC to the male EM. Similarly, Figure 17 shows the connections made from the headers on the TRF7970ATB to the onboard signals. The resulting mapping between the TIDC-EVSE-NFC and the TRF7970ATB is shown in . Each row in Table 4 represents a mapping between a pin of the EM header to the corresponding pin on the TRF7970ATB, where the highlighted rows represent the connections that are actually used for communication between the MSP430F6736 and the TRF7970A. Pins that do not have a mapping in are denoted by *N/C*.

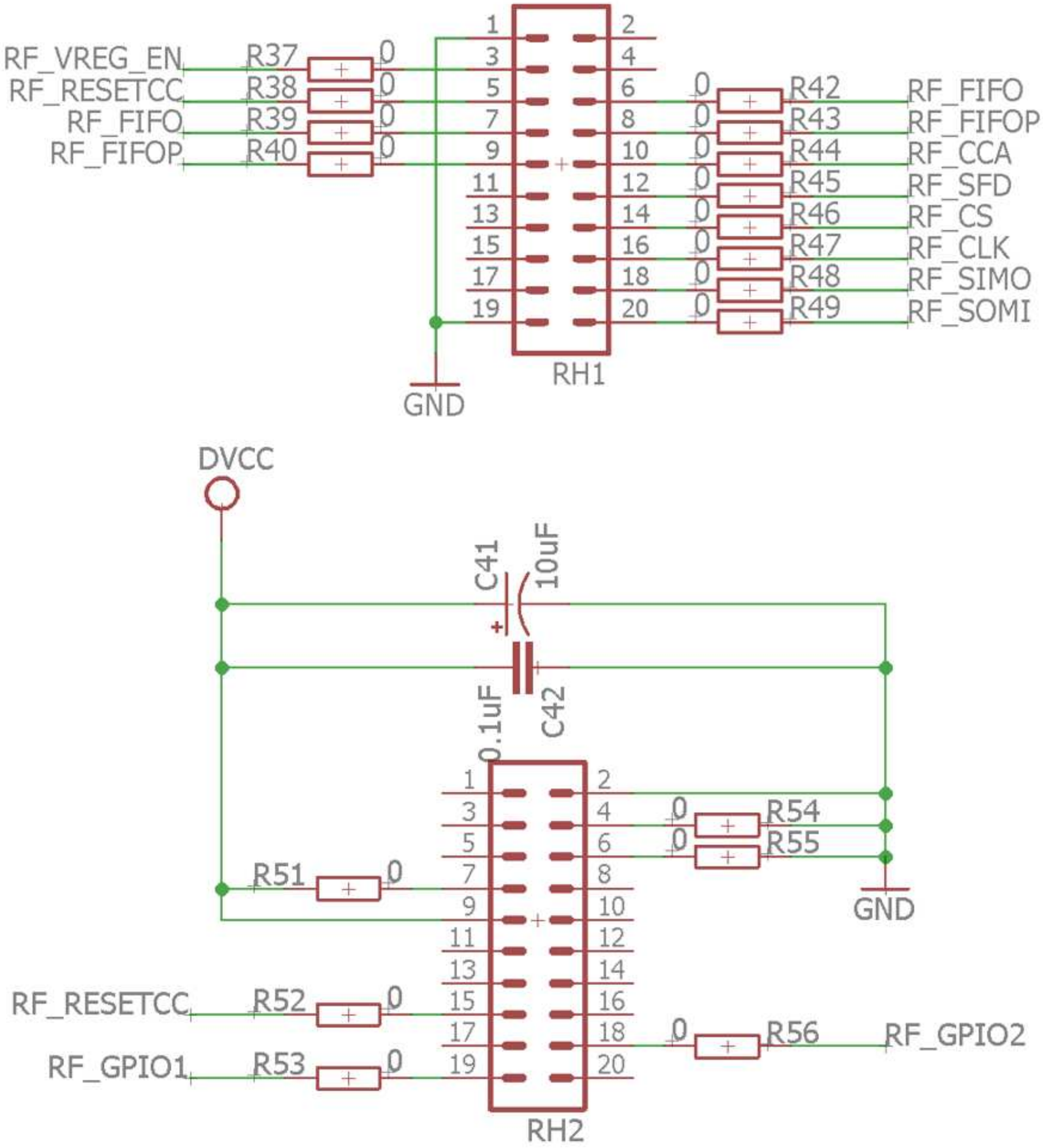


Figure 16. Connections on TIDC-EVSE-NFC RH1 and RH2 Connectors

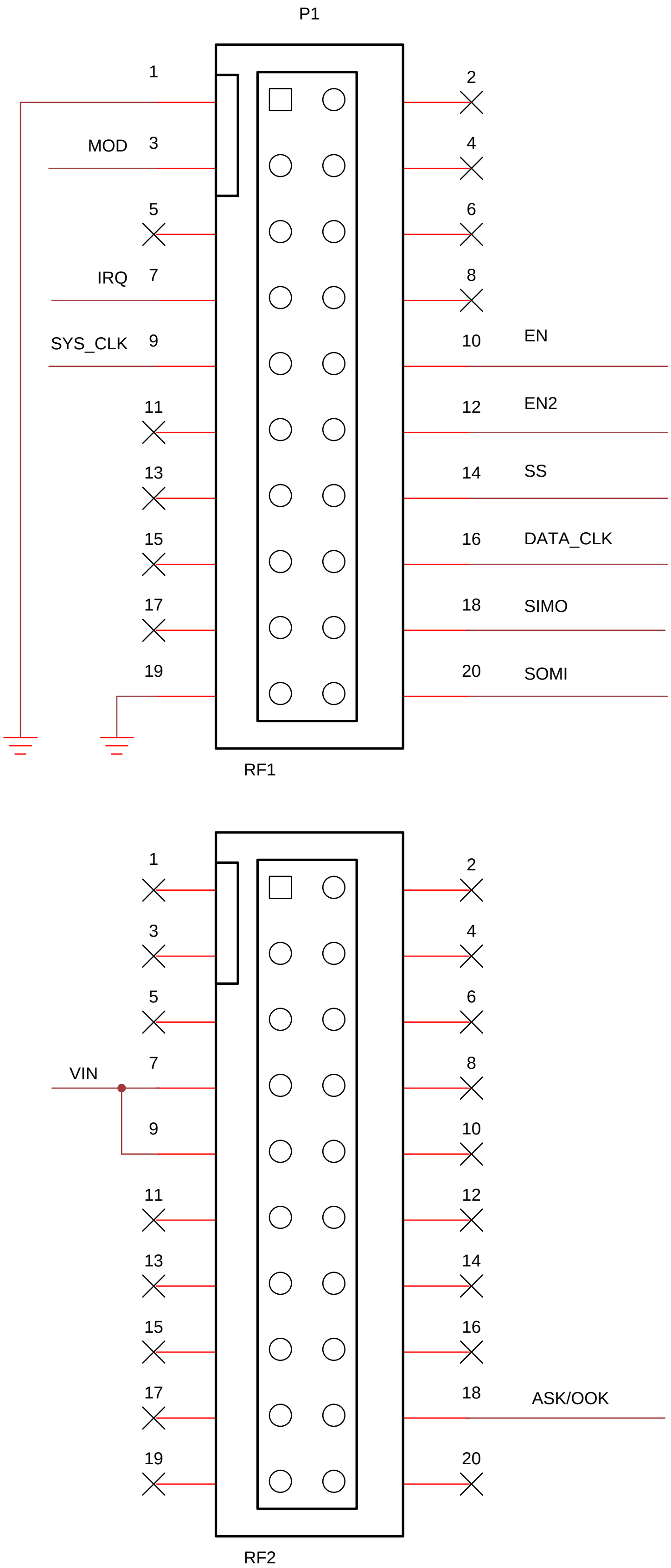


Figure 17. Connections on TRF7970ATB RF1 and RF2 Connectors

Table 4. Mapping Between the TIDC-EVSE-NFC and TRF7970ATB

TIDC-EVSE-NFC CONNECTION	TIDC-EVSE-NFC CONNECTION NAME	TRF7970ATB CONNECTION	TRF7970ATB CONNECTION NAME
RH1, pin 1	DGND	RF1, pin 1	GND
RH1, pin 3	RF_VREG_EN	RF1, pin 3	MOD
RH1, pin 5	RF_RESETCC	RF1, pin 5	N/C
RH1, pin 7	RF_FIFO	RF1, pin 7	IRQ
RH1, pin 9	RF_FIFOP	RF1, pin 9	SYS_CLK
RH1, pin 11	N/C	RF1, pin 11	N/C
RH1, pin 13	N/C	RF1, pin 13	N/C
RH1, pin 15	N/C	RF1, pin 15	N/C
RH1, pin 17	N/C	RF1, pin 17	N/C
RH1, pin 19	DGND	RF1, pin 19	GND
RH1, pin 2	N/C	RF1, pin 2	N/C
RH1, pin 4	N/C	RF1, pin 4	N/C
RH1, pin 6	RF_FIFO	RF1, pin 6	N/C
RH1, pin 8	RF_FIFOP	RF1, pin 8	N/C
RH1, pin 10	RF_CCA	RF1, pin 10	ENABLE
RH1, pin 12	RF_SFD	RF1, pin 12	ENABLE2
RH1, pin 14	RF_CS	RF1, pin 14	SLAVE_SELECT
RH1, pin 16	RF_CLK	RF1, pin 16	DATA_CLK
RH1, pin 18	RF_SIMO	RF1, pin 18	SIMO
RH1, pin 20	RF_SOMI	RF1, pin 20	SOMI
RH2, pin 1	N/C	RF2, pin 1	N/C
RH2, pin 3	N/C	RF2, pin 3	N/C
RH2, pin 5	N/C	RF2, pin 5	N/C
RH2, pin 7	DVCC	RF2, pin 7	+3.3VDC_IN
RH2, pin 9	DVCC	RF2, pin 9	+3.3VDC_IN
RH2 pin 11	N/C	RF2, pin 11	N/C
RH2, pin 13	N/C	RF2, pin 13	N/C
RH2, pin 15	RF_RESETCC	RF2, pin 15	N/C
RH2, pin 17	N/C	RF2, pin 17	N/C
RH2, pin 19	RF_GPIO1	RF2, pin 19	N/C
RH2, pin 2	DGND	RF2, pin 2	N/C
RH2, pin 4	DGND	RF2, pin 4	N/C
RH2, pin 6	DGND	RF2, pin 6	N/C
RH2, pin 8	N/C	RF2, pin 8	N/C
RH2, pin 10	N/C	RF2, pin 10	N/C
RH2, pin 12	N/C	RF2, pin 12	N/C
RH2, pin 14	N/C	RF2, pin 14	N/C
RH2, pin 16	N/C	RF2, pin 16	N/C
RH2, pin 18	RF_GPIO2	RF2, pin 18	ASK/OOK
RH2, pin 20	N/C	RF2, pin 20	N/C

5 Getting Started Firmware

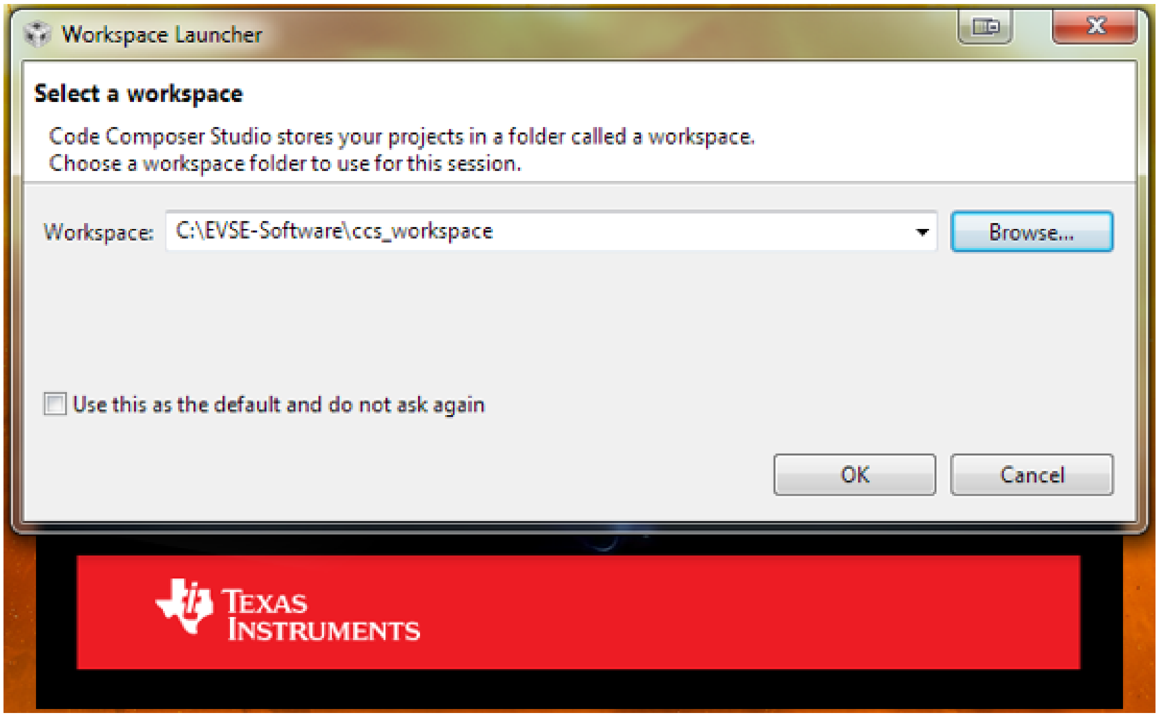
The software provided with this TI Design has been used to integrate the NFCLink library with the core EVSE functionality from TIDA-00637. This includes the J1772 signaling and response, a basic state machine for dealing with the signaling protocol, and has simple tag authentication added on top. This software is not production ready but shows the basic principles of integrating the two feature sets which can be leveraged to build the final application.

The software was built on top of the existing MSP430-Energy-Library. This library has been proven for use in electricity meters and provides an excellent framework on which to add the EVSE application and NFCLink state machines.

5.1 Workspace Setup

The software provided in this design requires the latest version of the Code Composer Studio™ (CCS) IDE from TI with the MSP430 plugins.

When extracting the software, ensure that all of the directory structures remain unchanged. When opening CCS (or switching workspaces), select the *ccs_workspace* directory that was extracted from the archive. The following example in shows placement of the files in the C:\EVSE-Software\ directory.



Workspace Launcher

Select a workspace

Code Composer Studio stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\EVSE-Software\ccs_workspace

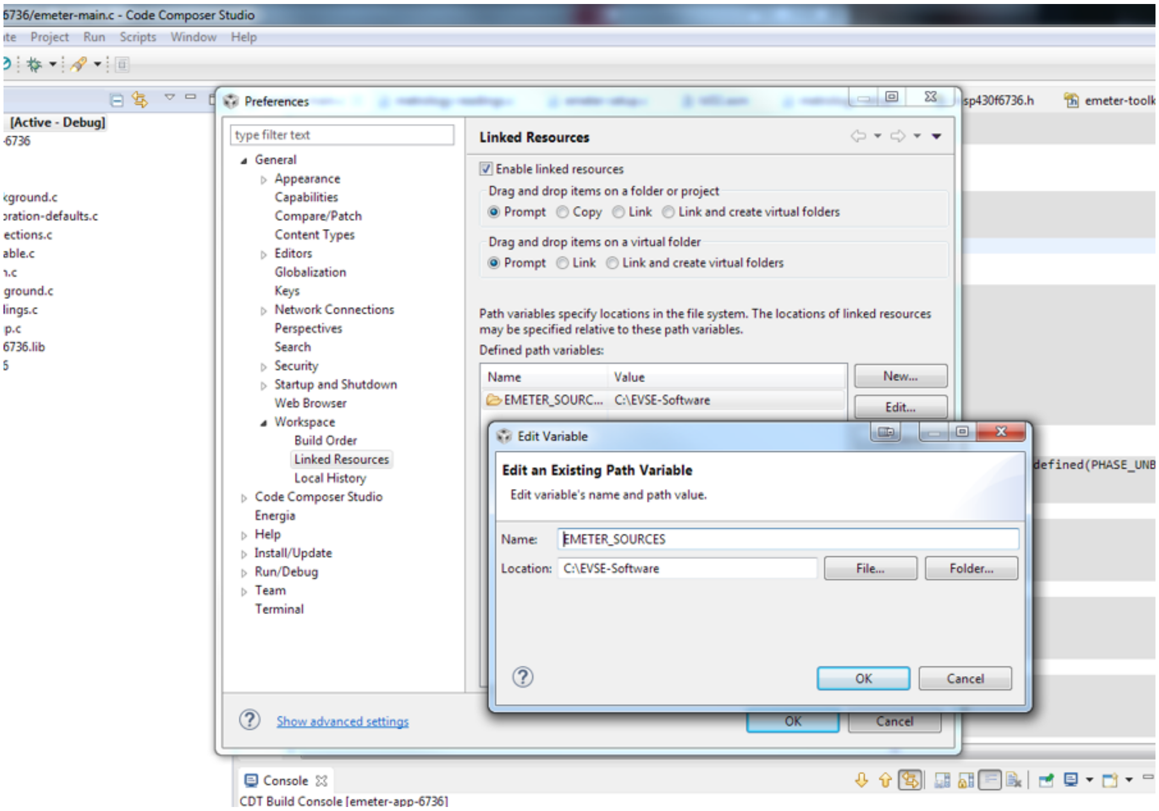
□ Use this as the default and do not ask again

Figure 18. Opening the CCS Workspace

Upon launching the workspace, three projects become visible:

1. **Emeter-app-6736**: This project contains the application level code, including peripheral setup and foreground state machines, as well as the driver for the TRF7970A.
2. **Emeter-metrology-6736**: This project is located one level below and has all of the metrology functionality. This functionality includes the ADC ISRs, metrology DSP calculations, and data access routines.
3. **Emeter-toolkit-6736**: This project has many low-level functions that are used to accelerate processing for specific data types used in the metrology processing engine.

When opening the workspace for the first time, the user is required to set a global variable to reference files. Set this variable by navigating to Window → Preferences in the File menu on the task bar. When the preferences window appears, navigate through the sidebar to General → Workspace → Linked Resources. In the Linked Resources tab, edit the resource EMETER_SOURCES to be one directory above the ccs_workspace directory. In this example, , that directory is C:\EVSE-Software\.



Linked Resources

☑ Enable linked resources

Drag and drop items on a folder or project
● Prompt ○ Copy ○ Link ○ Link and create virtual folders

Drag and drop items on a virtual folder
● Prompt ○ Link ○ Link and create virtual folders

Path variables specify locations in the file system. The locations of linked resources may be specified relative to these path variables.

Defined path variables:

Name	Value
EMETER_SOURC...	C:\EVSE-Software

Edit an Existing Path Variable

Edit variable's name and path value.

Name: EMETER_SOURCES

Location: C:\EVSE-Software

Figure 19. Changing EMETER_SOURCES Resource

The directory change can be tested by opening one of the C:\ drive files and recompiling the projects in the workspace.

When browsing the source code, major changes to the base library for the added NFC support are flagged with the following comment block, which allows the user to quickly find and discern the core functionality:

/* --------------------------- */

/* NFC Specific functionality */

/* --------------------------- */

In addition, the supporting library files are included within the emeter-app-F6736/nfc and emeter-app-F6736/nfc_gui_statemachines folders. The first contains the low lever device access drivers, and the second has sample state machines for processing various NFC tag types.

The firmware can be flashed as is to evaluate functionality by initiating a debug of the emeter-app-6736 project.

Any changes made to the individual projects must be propagated by recompiling the higher level projects because the emeter-toolkit-6736 is used by emeter-metrology-6736, which is in turn used by emeter-app-6736.

5.2 Energy Measurement

The energy measurement functionality of the TIDC-EVSE-NFC reference design is based on the MSP430-Energy-Library. Consult the documentation regarding the class 0.2 single-phase e-meter (TIDM-SINGLEPHASEEMETER-AFE).

5.3 Hardware Specific Setup

All pins on the MSP430 device are set at the application launch based on the functionality defined in emeter-template.h file. The three core registers required to initialize a pin for each I/O module (PxDIR, PxSEL, and PxOUT) are defined here and applied in the system_setup() register. All clocking and power options have been left on the default settings, the same as defined in the base MSP430-Energy-Library. Based on the reference design, the only additional setup required that is not covered by basic I/O settings is the pilot signal generation, GFCI interrupt, and ADC10 configuration.

5.3.1 Pilot Signal Setup

Because the pilot signal requires a steady 1-kHz PWM, the hardware has been designed to utilize a built-in timer module on the MSP430 device, known as TA2.1. The standard method of PWM generation on the MSP430 is to use one timer capture register to set the PWM frequency and another (with the appropriate output tied to a pin) as the PWM duty cycle.

The existing energy library clocking schema sets the SMCLK pin to match the MCLK pin at 25.16 MHz. Using an SMCLK with a divider of 25160 gives the appropriate timer frequency of 1 kHz. The PWM duty cycle can be set once and left alone because it is static through the operation of the EVSE (because it is based on the service connection and electromechanical design). Using Equation 1 and Equation 2, the user can determine the value for the TA2.1 trigger register as a percentage of 25160.

In the provided source code, the maximum current value is defined in a header; however, this value can be set programmatically if a variable value is required.

The initial state of the pilot wire is 12 V, or simply a high output from the timer module. The software sets up the timer and runs it at start-up, but does not output the signal. The signal output transpires in the application state machine when required.

5.3.2 Pilot Signal Setup

A simple I/O interrupt is required to enable the GFCI checking. On the MSP430 device, only ports one and two are capable of acting as interrupts. In the emeter_setup() function, the user must be sure to enable the P1.6 interrupt because the default register settings of the energy library do not allow implementing this setting.

5.3.3 ADC10 Configuration

All of the analog setup for the energy library occurs in the emeter-metrology-6736 project, in the metrology-setup.c file. The energy library already has hooks in place to use the ADC10 device for energy measurement, which the user can repurpose to measure the pilot wire input. The SD24 interrupt for the metrology triggers at a 4-kHz rate, which is also sufficient for measuring the 1-kHz pilot signal.

The MSP430F6736 contains a port map feature that must be set to enable ADC10 input. This feature is unique to the MSP430F67xx family and not required if the device is changed.

The ADC10 has been configured with fairly standard settings, but has also been set to require an external trigger. This trigger runs is set in the SD24 ISR so that the ADC10 samples as soon as it finishes with the background DSP process.

5.4 NFCLink Stack

5.4.1 Software Overview

To add NFC capabilities to the EVSE platform software, the TRF7970 is provided with the NFCLink software stack. This stack can be integrated into any application to add a variety of NFC standard functions. In this case, the library will be modified to support reading a type 5 tag, but any other types can be easily enabled.

The library comes with two folders that break up the functional. /nfc contains the device drivers, including SPI access, timer functions, and message formatting; while /nfc_gui_statemachines includes the various methods of reading and writing the different tag types. For the purposes of this design on the t5 state machine is used.

Some of the files in the nfc directory have been modified to support the EVSE platform hardware used as a base in this design, but most of the library remains as shipped and is device agnostic. In addition, several files from the MSP430 driverlib have been included as needed to support the software architecture used in NFCLib. lists these files and an overview of their functionality and changes.

Table 5. TIDC-EVSE-NFC NFCLink Source Files

FILENAME	DESCRIPTION
eusci_a_spi.c	Driver for the eusci_a_spi Module. Unchanged from MSP430 Driver Lib.
gpio.c	Driver for the gpio Module. Unchanged from MSP430 Driver Lib.
iso_7816_4.c	Implementation of ISO7816-4 APIs. Unchanged from NFCLink.
iso_dep.c	Implementation of ISO DEP APIs. Unchanged from NFCLink.
llcp.c	Logic Link Control Protocol : used to establish / mantain a link to send packets via NPP or SNEP. Unchanged from NFCLink.
mcu.c	MCU Configuration and host interface APIs. Configuration changed from original source to include correct clocking parameters for Timer
nfc_a.c	Implementation of ISO14443-A APIs. Unchanged from NFCLink.
nfc_b.c	Implementation of ISO14443-B APIs. Unchanged from NFCLink.
nfc_controller.c	Implementation of NFC Controller APIs. Unchanged from NFCLink.
nfc_dep.c	Used to send packets in P2P. Unchanged from NFCLink.
nfc_f.c	Contains implementation of NFC Type F (Felica) protocol. Unchanged from NFCLink.
nfc_initiator.c	Implementation of the polling mode. Unchanged from NFCLink.
nfc_rw_t2t.c	Implementation of T2T Reader/Writer. Unchanged from NFCLink.
nfc_rw_t3t.c	Implementation of T3T Reader/Writer. Unchanged from NFCLink.
nfc_rw_t5t.c	Implementation of T5T Reader/Writer. Unchanged from NFCLink.
nfc_spi.c	SPI Configuration and transmission APIs. Changed to use UCA2 rather than UCB1, with updated clock information.
nfc_target.c	Implementation of the target/listen mode. Unchanged from NFCLink.
pmm.c	Driver for the pmm Module. Unchanged from MSP430 Driver Lib.
sfr.c	Driver for the sfr Module. Unchanged from MSP430 Driver Lib.
snep.c	Implementation of Simple NDEF Exchange Protocol, used by LLCP and the main application. Unchanged from NFCLink.
timer_a.c	Driver for the timer_a Module. Unchanged from MSP430 Driver Lib.
trf7970.c	TRF7970A Firmware Driver. Unchanged from NFCLink.
ucs.c	Driver for the ucs Module. Unchanged from MSP430 Driver Lib.
wdt_a.c	Driver for the wdt_a Module. Unchanged from MSP430 Driver Lib.

Each file in the newly included also has an associated header file, but there are several which define some global settings. These are outlined in Table 6.

Table 6. TIDC-EVSE-NFC NFCLink Header Files

FILENAME	DESCRIPTION
nfc_config.h	NFC Command Definitions. Unused modes and some debug functions have been disabled.
mcu.h	MCU Configuration and host interface APIs definitions. Clocks were defined to enable access by software resources, but these do not define the system clocks which are set in the emeter-application.
config.h	MCU Pin Configuration to interface with the TRF7970A. The pinmap was changed to reflect the hardware in place.

5.4.2 Start-up Sequence

After the TIDC-EVSE-NFC is powered and its peripherals are configured, the TRF7970A and associated software are configured for operation by the EVM. These configuration settings are defined in *NFC_configuration()* located in emeter-main.c. By default, this implementation is only supporting iso15693 at 48 kbps. This configuration will need to be altered to reflect the specific tag technology used.

At this stage the TRF7970A will be in a polling mode, and the EVSE functional will be idle. The software will continuously look for a newly presented tag and alert the MSP430 to its presence through the IRQ line. When a card is presented, the software will be able to begin reading specific blocks of the card.

5.5 EVSE Application State Machine

The core functionality of the EVSE is primarily moving between different states of the J1772 pilot signal protocol. The simplest method of facilitating this movement is by implementing a simple state machine that has been built in the foreground of the energy library. The energy library has a function in the application layer that runs once per second (based on the number of SD24 samples collected), which provides the basis for the state machine. shows the basic layout of the implemented state machine .

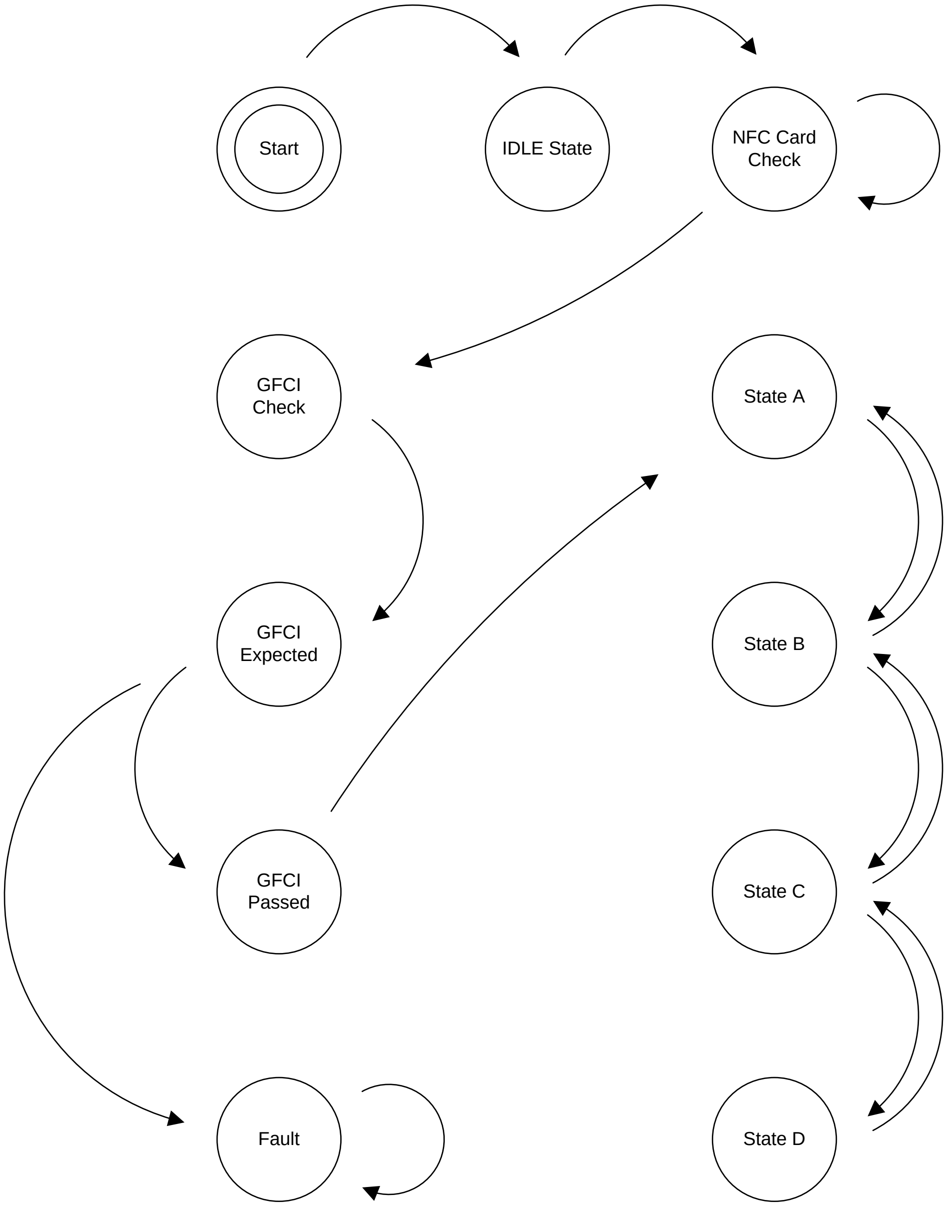


Figure 20. State Machine Overview

The following list outlines the process of the implemented state machine:

1. **Start**: State machine entry point. Any additional initialization can be put here.
2. **IDLE State**: Post initialization. The EVSE system enters an idle state until it is authorised to enable the charging state machine.
3. **NFC Card Check**: While idle, the system will continuously monitor for the presence of an NFC card. Upon its presentation, the system will authenticate the card through a local or remote solution. If the card is valid, the system moves to running its internal self check, otherwise, the stystem will remain in the idle state.

4. **GFCI Check**: This state initializes a GFCI check by setting a flag for the ISR and outputting a pulse to the check coil. The pulse triggers the GFCI ISR, which does not trigger a fault if the check expected flag has been set.
5. **GFCI Expected**: Check that the GFCI test has been properly detected here. Trigger a GFCI Check fault and move to the fault state if necessary. If passed, move to the **GFCI Passed** state.
6. **GFCI Passed**: The pass state is a mostly empty state to facilitate any additional functionality that may be required to run before starting the J1772 signaling.
7. **State A**: To start the signaling, the pilot line is brought high at the beginning of State A. The only possible exit from here is to State B when the voltage drop on the pilot line has been signaled by the ADC10 ISR in the background. If a different voltage has been detected, this state can exit to the Fault state with a J1772 condition.
8. **State B**: State B enables the PWM output on the pilot line. From here, the only possible exit is to State C (indicating that the vehicle has changed to resistance to signal that it is ready to accept charge voltage), State A (indicating that the connector has been unplugged), or a J1772 fault condition that has been detected in the background.
9. **State C**: State C enables the charge voltage by closing the relay. Possible exits include State B (indicating that the vehicle has changed the resistance to signal that it is done charging), State A (indicating that the connector has been unplugged from the vehicle), State D (indicating that venting is required), or a J1772 fault condition that has been detected in the background.
10. **State D**: Indicates that venting is required. This state is implemented in software, but only returns to State C or to the fault condition.
11. **Fault**: The primary faults that can be triggered by the system are GFCI check failed, GFCI triggered, relay stuck, or a J1772 pilot signal fault. These faults are all critical and require a system restart to clear.

The primary mechanism for shifting between various states is the pilot wire measurement on the ADC10 module. When a voltage change has been measured, the foreground state machine detects this change and moves appropriately. This setup limits the response rate to 1 s because the foreground operates at 1 Hz, which is a sufficiently fast response rate for the application at hand.

The "NFC Card Check" state is where the NFCLink software stack integration has been implemented for this design. The application will call into the library at every tick of the main loop using the function NFC_run(). This returns the current state of the underlying NFC driver. Based on the current state, the appropriate lower layer state machine is used to process the card. In order to perform a full read of a card, several steps are performed, each at a tick of the main application:

1. Card Inventory
2. Info Status Check
3. Get Info Blocks
4. Read if NDEF or RAW
5. Read Data (performed until all data is read)

Based on this process, the NFCLink library must be called into many times to read the card data, but this allows the parent application to maintain as much control as possible during this process. When the read is fully complete, the library state machines are set to idle and a callback can inform the parent application that data is available to be authenticated against.

Additional information regarding the lower level operation of the NFCLink library can be obtained in TIDesign TIDM-NFC-RW.

6 Test Setup

The test setup used for TIDC-EVSE-NFC mimics that of TIDA-00637. The only alteration is the addition of the TRF7970ATB as described in Section 4.

7 Test Data

The core functionality of the EVSE was tested in TIDA-00637. The test data for this design will be focusing on the NFC tag reading.

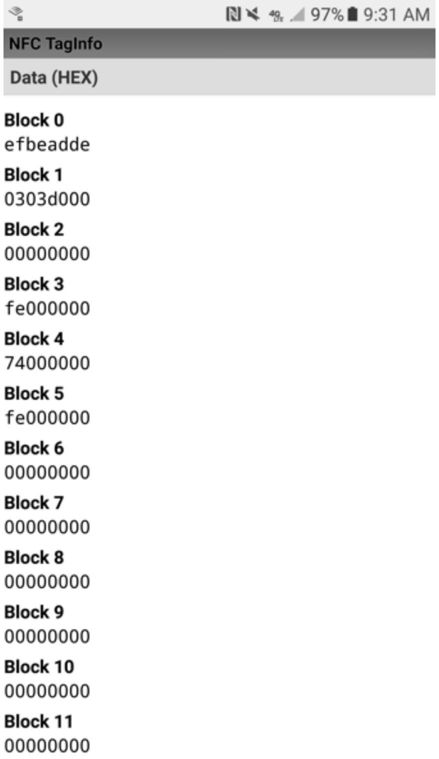
7.1 NFC Read and Validation

The TRF7970ATB daughter card comes with two tags that have some data already written to them, an ISO14443 MIFARE tag, and an ISO15693 tag. For this test the ISO15693 tag will be used. A simple block of data will be written to the tag to serve as the authentication mechanism. In this test block zero of the tag has been set to 0xBEEFDEAD using the TRF7970AEVM, which will serve as the check. Any data will suffice for this example but having something recognizable will assist in debugging. This tag type falls into the T5T category in NFC_Link, so only that layer has been modified to serve the purposes of this design.

In this simple implementation, the NFC_Link software copies the data from all 64 blocks on the tag into a memory bank that can be read from the host application. Upon a successful read of a tag, the application layer is notified that the data is available (through the T5T_readStatus() function call) , and the application layer is able to run its own check for validity.

For this example, a sample validity check function has been written, named ProcessNFCData(). This function is called when data is successfully read and can be copied into a memory block. The function runs a simple byte check for a single known good card. In a more full featured application, this check can be replaced with an external authentication mechanism.

The data from the example ISO15693 card can be read with a simple Android application. In this case the *NFC TagInfo* is used, which is able to read the raw data. This data is validated in. The formatting is slightly changed due to the over-the-air formatting of NDEF messages, but the data is still accurate.



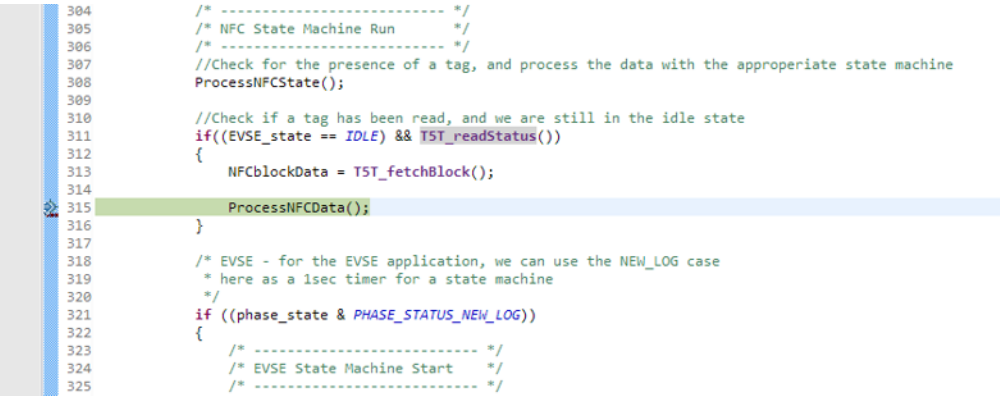
```
Block 0
efbeadde
Block 1
0303d000
Block 2
00000000
Block 3
fe000000
Block 4
74000000
Block 5
fe000000
Block 6
00000000
Block 7
00000000
Block 8
00000000
Block 9
00000000
Block 10
00000000
Block 11
00000000
```

Figure 21. Raw Data Read From NFC Tag

A few steps must be followed to setup the test environment:

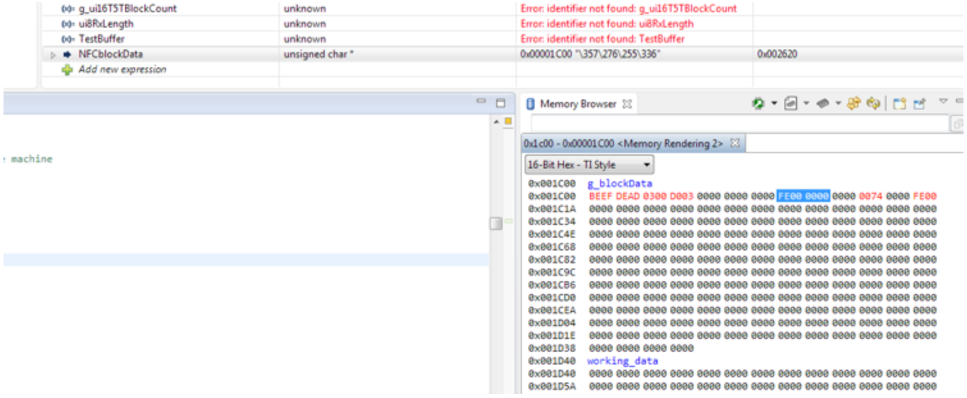
1. Setup TIDA-00637 with an MSP debugger, TRF7970ATB, and power.
2. Launch the CCS workspace containing the sample application.
3. Enable a breakpoint in emeter-main.c at line 315.
4. Enable a breakpoint in emeter-main.c at line 1182.
5. Launch debugger.

To test the card reading place several breakpoints in the software to check the data coming from the TRF7970ATB against expected results. The first is at line 315, where the application has detected that a new tag has been read, and data is available. When the tag has been read, the software will stop here so a data check can be performed, as shown in and .



```
304         /* -------------------------- */
305         /* NFC State Machine Run      */
306         /* -------------------------- */
307         //Check for the presence of a tag, and process the data with the approperiate state machine
308         ProcessNFCState();
309
310         //Check if a tag has been read, and we are still in the idle state
311         if((EVSE_state == IDLE) && T5T_readStatus())
312         {
313             NFCblockData = T5T_fetchBlock();
314
315             ProcessNFCData();
316         }
317
318         /* EVSE - for the EVSE application, we can use the NEW_LOG case
319          * here as a 1sec timer for a state machine
320          */
321         if ((phase_state & PHASE_STATUS_NEW_LOG))
322         {
323             /* -------------------------- */
324             /* EVSE State Machine Start   */
325             /* -------------------------- */
```

Figure 22. Break Point When Tag Data is Available

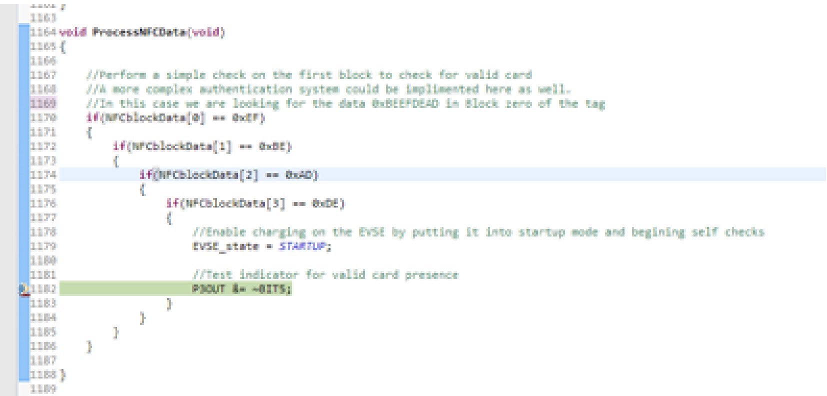


g_ui16T5TBlockCount	unknown	Error: identifier not found: g_ui16T5TBlockCount	
ui8RxLength	unknown	Error: identifier not found: ui8RxLength	
TestBuffer	unknown	Error: identifier not found: TestBuffer	
NFCblockData	unsigned char *	0x00001C00 "\357\276\255\336"	0x002620
Add new expression			

```
0x001C00  g_blockData
0x001C00  BEEF DEAD 0300 D003 0000 0000 0000 FE00 0000 0000 0074 0000 FE00
0x001C1A  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0x001C34  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0x001C4E  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0x001C68  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0x001C82  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0x001C9C  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0x001CB6  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0x001CD0  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0x001CEA  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0x001D04  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0x001D1E  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0x001D38  0000 0000 0000 0000
0x001D40  working_data
0x001D40  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0x001D5A  0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
```

Figure 23. Tag Data Observation in Memory Explorer

The data read from the card for the foreground is located at the pointer NFCblockData, which is mapping to a location in the background called g_blockData. If viewing the memory at that location in the memory browser, it cab be observed that the memory data matches the raw data we captured from the card earlier. The authentication mechanism in the example is using simple byte checking implemented in ProcessNFCData, shown in . Once fully authenticated the check will enable the EVSE and set an external indicator. A more fully-featured mechanism can be implemented by the end user.



```
void ProcessNFCData(void)
{

    //Perform a simple check on the first block to check for valid card
    //A more complex authentication system could be implimented here as well.
    //In this case we are looking for the data 0xBEEFDEAD in Block zero of the tag
    if(NFCblockData[0] == 0xEF)
    {
        if(NFCblockData[1] == 0xBE)
        {
            if(NFCblockData[2] == 0xAD)
            {
                if(NFCblockData[3] == 0xDE)
                {
                    //Enable charging on the EVSE by putting it into startup mode and begining self checks
                    EVSE_state = STARTUP;

                    //Test indicator for valid card presence
                    P3OUT &= ~BIT5;
                }
            }
        }
    }

}
```

Figure 24. Simple Data Validation by Application

7.2 Energy Measurement

The energy measurement accuracy results for this reference design are identical to the results in the class 0.2 single-phase e-meter design. Reference this design for a full analysis of results (TIDM-SINGLEPHASEMETER-AFE).

7.3 Power Supply

For a full analysis of the power supply, upon which the TIDA-00637 design was based, visit the PMP10299.1 tool folder (PMP10299.1)

8 Design Files

8.1 Schematics

To download the schematics, see the design files at TIDC-EVSE-NFC.

8.2 Bill of Materials

To download the bill of materials (BOM), see the design files at TIDC-EVSE-NFC.

8.3 Layer Plots

To download the layer plots, see the design files at TIDC-EVSE-NFC.

8.4 CAD Project Files

To download the CAD project files, see the design files at TIDC-EVSE-NFC.

8.5 Gerber Files

To download the Gerber files, see the design files at TIDC-EVSE-NFC.

8.6 Software Files

To download the software files, see the design files at TIDC-EVSE-NFC.

9 Related Documentation

1. Texas Instruments, *Class 0.2 Single-Phase E-Meter*, Application Note (TIDUBV2)
2. Texas Instruments, *NFC Transceiver Add-on Target Board Module*, TRF7970ATB Tools Folder
3. Texas Instruments, *TRF79x0ATB NFC/HF RFID Reader Module*, Application Note (SLOU372)
4. Texas Instruments, *Level 1&2 Electric Vehicle Service Equipment Reference Design*, TIDA-00637 Design Folder
5. Texas Instruments, Near Field Communication (NFC) Reader/Writer Reference Design, TIDM-NFC-RW Design Folder
6. Texas Instruments, *SPICE-Based Analog Simulation Program*, TINA-TI Tools Folder

9.1 Trademarks

10 Terminology

EVSE – Electric Vehicle Service Equipment

EV – Electric Vehicle

Pilot Line and Pilot Wire – One-wire communication between an EVSE and an EV

NFC – Near Field Communication for short range, passive, wireless data transfer

11 About the Author

BART BASILEis a systems architect in the Grid Infrastructure Solutions Team at Texas Instruments, focusing on renewable energy and EV infrastructure. Bart works across multiple product families and technologies to leverage the best solutions possible for system level application designs. Bart received his Bachelor's of Science in Electronics Engineering from Texas A&M University.

IMPORTANT NOTICE FOR TI REFERENCE DESIGNS

Texas Instruments Incorporated (‘TI”) reference designs are solely intended to assist designers (“Designer(s)”) who are developing systems that incorporate TI products. TI has not conducted any testing other than that specifically described in the published documentation for a particular reference design.

TI’s provision of reference designs and any other technical, applications or design advice, quality characterization, reliability data or other information or services does not expand or otherwise alter TI’s applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such reference designs or other items.

TI reserves the right to make corrections, enhancements, improvements and other changes to its reference designs and other items.

Designer understands and agrees that Designer remains responsible for using its independent analysis, evaluation and judgment in designing Designer’s systems and products, and has full and exclusive responsibility to assure the safety of its products and compliance of its products (and of all TI products used in or for such Designer’s products) with all applicable regulations, laws and other applicable requirements. Designer represents that, with respect to its applications, it has all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. Designer agrees that prior to using or distributing any systems that include TI products, Designer will thoroughly test such systems and the functionality of such TI products as used in such systems. Designer may not use any TI products in life-critical medical equipment unless authorized officers of the parties have executed a special contract specifically governing such use. Life-critical medical equipment is medical equipment where failure of such equipment would cause serious bodily injury or death (e.g., life support, pacemakers, defibrillators, heart pumps, neurostimulators, and implantables). Such equipment includes, without limitation, all medical devices identified by the U.S. Food and Drug Administration as Class III devices and equivalent classifications outside the U.S.

Designers are authorized to use, copy and modify any individual TI reference design only in connection with the development of end products that include the TI product(s) identified in that reference design. HOWEVER, NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of the reference design or other items described above may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI REFERENCE DESIGNS AND OTHER ITEMS DESCRIBED ABOVE ARE PROVIDED “AS IS” AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING THE REFERENCE DESIGNS OR USE OF THE REFERENCE DESIGNS, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY DESIGNERS AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS AS DESCRIBED IN A TI REFERENCE DESIGN OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF THE REFERENCE DESIGNS OR USE OF THE REFERENCE DESIGNS, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

TI’s standard terms of sale for semiconductor products (http://www.ti.com/sc/docs/stdterms.htm) apply to the sale of packaged integrated circuit products. Additional terms may apply to the use or sale of other types of TI products and services.

Designer will fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of Designer’s non-compliance with the terms and provisions of this Notice.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265