Accepted to:
Seventh Global Conference on Power Control and Optimization PCO 2013

DESIGN OF SMART CHARGING INFRASTRUCTURE HARDWARE AND FIRMWARE DESIGN OF THE VARIOUS CURRENT MULTIPLEXING CHARGING SYSTEM

Ching-Yen Chung, Peter Chu, Rajit Gadh
Department of Mechanical and Aerospace Engineering
University of California, Los Angeles
Los Angeles
chingyenchung@ucla.edu, peterchu@ucla.edu, gadh@ucla.edu

Abstract

Currently, when Electric Vehicles (EVs) are charging, they only have the option to charge at a selected current or not charge. When during the day there is a power shortage, the charging infrastructure should have the options to either shut off the power to the charging stations or to lower the power to the EVs in order to satisfy the needs of the grid. There is a need for technology that controls the current being disbursed to these electric vehicles. This paper proposes a design for a smart charging infrastructure capable of providing power to several EVs from one circuit by multiplexing power and providing charge control. The smart charging infrastructure includes the server and the smart charging station. With this smart charging infrastructure, the shortage of energy in a local grid could be solved by our EV management system.

Keywords*:* Electrical vehicle charging, power distribution control, smart grids, Wireless LAN, wireless mesh network, Zigbee

1. Introduction

Electric Vehicles (EV) are becoming a choice for many people as the world is beginning to lower its dependence on fossil fuels. As EVs on the road increase, charging stations in both parking structures and garages will become more important for longer distance commuters. An available charging station may be a critical requirement to ensure the ability to finish the round trip and make it home. In order to satisfy the demand of EV charging requests, an EV charging management system needs to be implemented to handle the peak time EV charging, regardless of charging taking place in a parking garage or at home. Charging stations that service multiple vehicles simultaneously with a given infrastructure need to be implemented in order to meet the increasing demand for charging infrastructure. The station needs to share the circuit by rationing the available power in order to not overload the circuit by intelligently scheduling charging. In this paper, a software-based EV monitoring, control and management system, WINSmartEV™[1][2], which allows remote monitoring and variable current control of EV charging is presented. This smart charging infrastructure is capable of providing power to several EVs from one circuit with multiplexing and variable current controlled charging. By using this smart charging infrastructure to manage EV charge scheduling and current flow, energy shortage in local grids can be prevented.

Current commercial EV charging stations like Coulomb[3] and Blink[4] have their own proprietary networks to connect the charging stations in service. Coulomb provides a ChargePoint application programming interface (API) and an OpenCharge protocol for the developers. The current application uses their network to locate available charging stations for its users. Coulomb also has a ChargePoint application programming interface (API) and an OpenCharge protocol for the developers. The current application uses their network to locate available charging stations for its users. It is possible to build the smart charging system by using the existing network with this API and protocols when they are obtained. Other commercial charging stations, Leviton[5] and Clippercreek[6], simply provide basic charging stations without any network features. The simple charging stations could be the platforms for developers to implement their own network services for smart charging purposes. In [7][8][9][10], several charging algorithms and results are presented. However, none of them discussed how to achieve variable current and multiplexing control. In section 2, we propose a smart charging infrastructure which is capable of providing power to several EVs from one circuit with multiplexing and various charging current.

2. System architecture

The smart charging infrastructure, WINSmartEV™, is illustrated in Fig. 1.

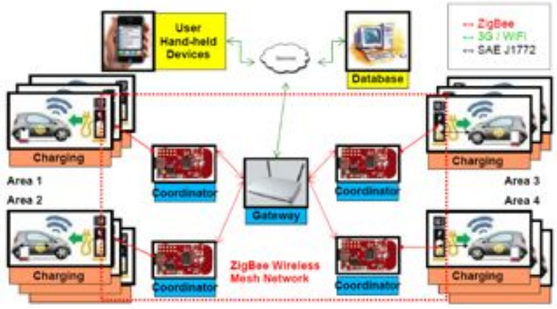

Fig. 1: Smart charging infrastructure

The server sends commands to the charging station and controls the charging speed while gathering and accumulating all the power information through the multiple protocol gateways with 3G communication. 3G communication is required due to its flexibility and accessibility to be everywhere as long as the cellular signal exists, especially where wired or WiFi communication is unavailable. Based on user preference and the local power capacity, the possible efficient control scheme is provided by sending the command to the charging station through a multiple protocol gateway. Since the National Institute of Standards and Technology (NIST) has announced the first draft of the framework and roadmap to coordinate the interoperability and standards for the smart grid[11], in which ZigBee is specified for its low power and mesh network capabilities, ZigBee has been adopted in our system. The details of the smart charging station are shown in Fig. 2

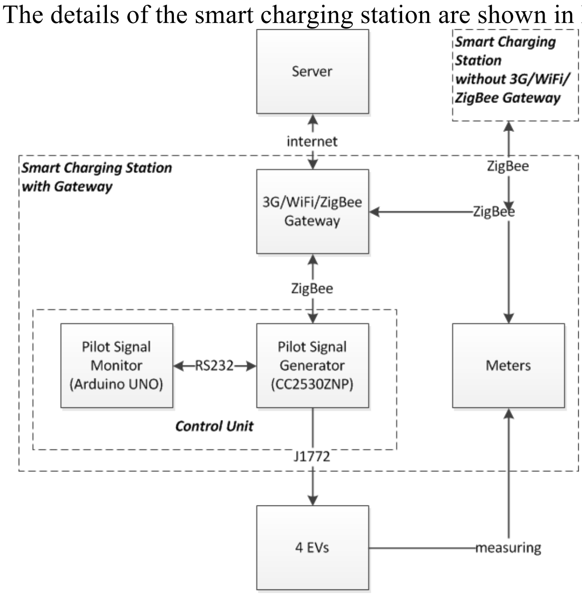

Fig. 2: Details of smart charging station

The information interchanged between the gateway, meters, and the control unit are through ZigBee communication. Because a number of the control devices in each charge station can communicate using ZigBee mesh network capabilities, the charging stations can communicate with each other. Therefore, only one gateway is required in each localized area to access the internet.

2.1. Metering system

Inside the charging station sit a gateway, 4 meters, 4 safety relays, and a control unit. Upon receiving the command of retrieving power information, the meter returns its power information, including voltage, current, and power, to the gateway through the ZigBee communication. The detail schematics of a four-channel metering system are shown in Fig. 3.

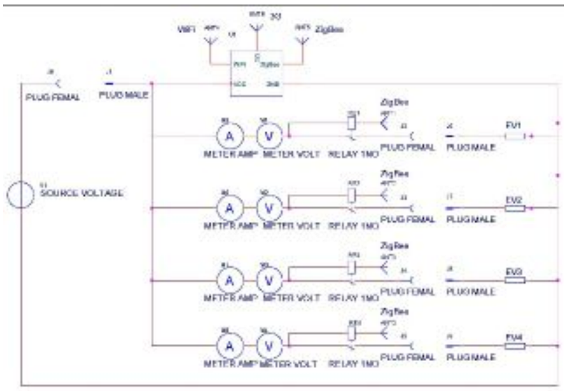

Fig. 3: Schematics of metering system

The relays of each channel can be turned on/off by the server, control unit of the charging station, or GFCI circuit. Fig. 4 shows the realization of the 4-channel smart charging station.

Fig. 4: Realization of 4 channel charging station

2.2. Control unit of the charging station

According to the standard of charging cable interface SAE J1772[12], the pilot signal is in Pulse Width Modulation (PWM) format. Based the duty cycle of the pilot signal provided by the charging station, the EV adjusts its load to meet the current limitation. With this characteristic, the charging station is able to control the charging speed by changing the duty cycle of the pilot signal. Fig. 5 graph shows the supply current rating versus the pilot signal duty cycle.

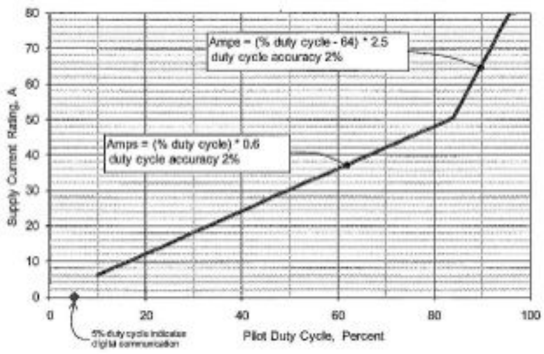

Fig. 5: Supply current v.s. pilot duty cycle[12]

2.2.1 Pilot signal generator and monitor

The pilot signal generator and monitor play the key role of the control unit in the smart charging station. With CC2530ZNP and Arduino UNO, the control unit including the pilot signal generator, pilot signal monitor, safety relay controller, and auto-reset function is implemented as shown in Fig. 6.

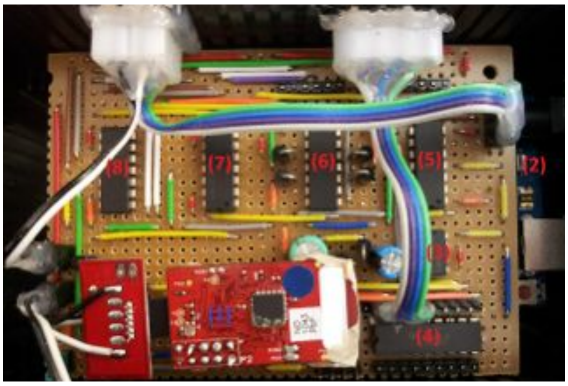

Fig. 6: Realization of the control unit: (1)pilot signal generator (2)pilot signal monitor (3)555 counter (4)relay driver (5)Schmitt trigger (6)unit gain buffer (7)inverting amplifier (8)inverting LPF

The pilot signal generator, which generates 4 PWM signals, is implemented in the MSP430 of CC2530ZNP as a ZigBee end device. The firmware flow of the MSP430 is shown in Fig. 7.

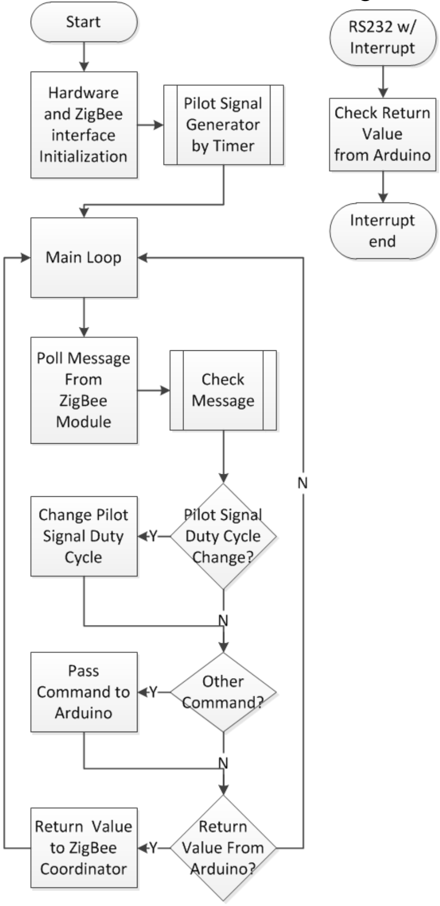

Fig. 7: Firmware flow of MSP430 of CC2530ZNP as the ZigBee end device (pilot signal generator)

In the firmware flow chart, after the initializing of the hardware and ZigBee interface, 4 PWM signals are generated by using two internal timers at the startup process. The message from the ZigBee chip, CC2530, is then polled in the main loop. Once the duty cycle change command message is checked, the duty cycle of the PWM will change in the main loop. Other commands will be passed to the Arduino.

According to J1772 specification[12], the EV plug-in status is detected with the pilot signal generated in the charging station. When there is no EV connected to the charging station, the voltage of the pilot signal pin on the handle should be DC +12V. After the user plugs-in the EV, the voltage of the pilot signal pin should be +9V or +6V, depending on whether the EV is ready to accept the energy or not. The charging station will start to generate the pilot signal with a certain duty cycle according to the power capacity of the charging station. When the EV is fully charged, the positive part of the pilot signal will be from +6V to +9V. When the user unplugs the

EV, the positive part of the pilot signal will be from +6V to +12V. Fig. 8 shows the firmware flow of the pilot signal monitor.

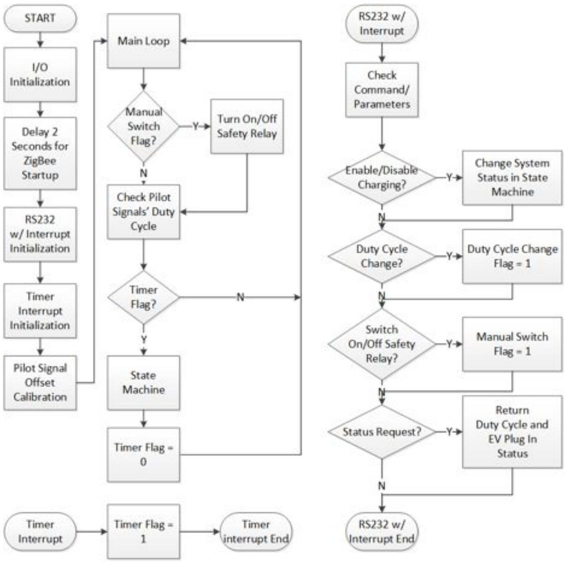

Fig. 8 Pilot signal monitor firmware flow chart

The message from MSP430 of CC2530ZNP is handled by RS232 with an interrupt loop; therefore, messages will not be missed while executing other process in the main loop. The actions including the commands check, flags set, and the message return are handled in the RS232 interrupt loop. Most of the actions are handled in the main loop according to the flags. With the timer interrupt, routine service such as checking for EV existence and turning the safety relay on/off, will have exact intervals. Moreover, without re-flashing the firmware, the interval of the routine service can be changed remotely, which makes the service more flexible. In the timer interrupt routine, instead of handling services, only the timer flag is handled. The reason for this is that, if we handle services in the timer interrupt routine, the required time of the timer interrupt routine will increase. When a RS232 command comes around the same time, either the command will be missed or the timer interrupt routine will be broken. Most of the unexpected actions are caused by this kind of unexpected firmware flow; therefore, the time of the timer interrupt routine should be as small as possible. It is flexible to add services based on this kind of firmware structure, since the firmware handle services according to the flags in the main loop.

The EV plug-in status detection is executed in the state machine of EV charging procedure. The detection of EV plug-in status is handled with the Timer Flag, which is set to be 1 in the timer interrupt loop. After the detection process is finished, the Timer Flag will be set to be 0. Currently, the interval of the timer interrupt is 1 second in the firmware, which means the detection process is handled every 1 second. Note that the detection process time needs to be less than the interval of the timer interrupt so that the detection process can be handled correctly. The RS232 with internal interrupt and timer interrupt are initialized at the startup process. After that, the pilot signal offset is calibrated before the EV detection. In the startup, the delay of two seconds before the initialization of a serial connection is required to wait for the CC2530 startup. Fig. 9 shows the firmware flow of the state machine.

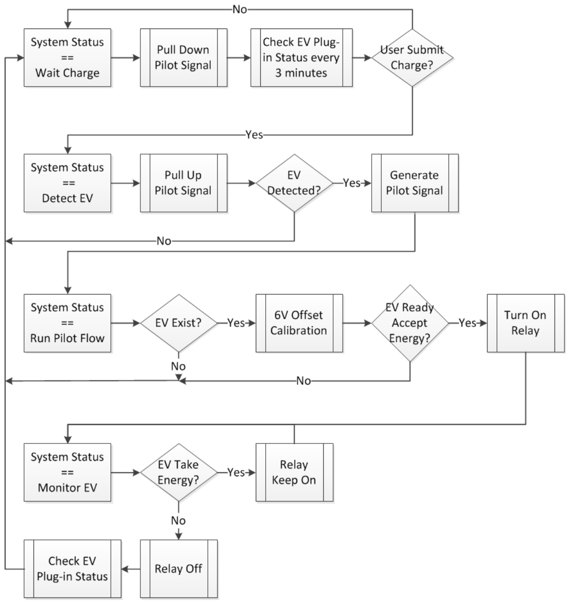

Fig. 9: State machine flow chart

As for the peripheral circuits, four 4-channel amplifiers are used to implement the 4-channel pilot signal generator and monitor, as shown in Fig. 10.

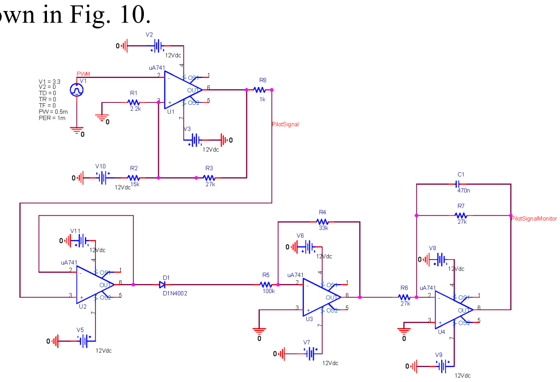

Fig. 10: Peripheral circuit simulation schematics

The PWM signal generated by the pilot signal generator is amplified from 3.3V/GND to +/-12V by the Schmitt trigger and fed to the EV. Following the Schmitt trigger is the unit gain buffer for separating the circuits so that the pilot signal monitor can monitor the pilot signal without adding load effect to the pilot signal. Note that a Schottky diode is inserted after the unit gain buffer because only the positive part of the signal will be monitored. To shrink the pilot signal to the A/D conversion range, an inverting amplifier with 1/3 gain is used. The peak value detection method is not utilized because it tends to be affected by unexpected spikes which make the measurement unreliable. Instead, an inverting active low pass filter (LPF) is used to average out the pilot signal such that a DC voltage can be measured by the pilot signal monitor. The peripheral circuits are simulated by PSpice as shown in Fig. 11.

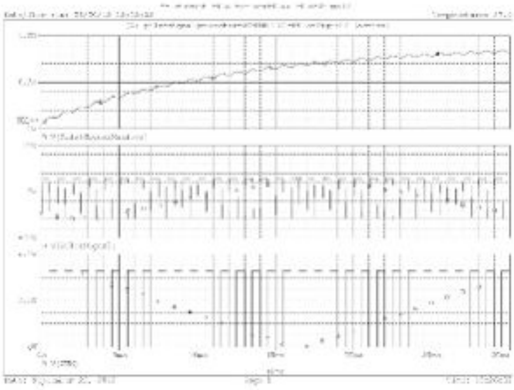

Fig. 11: Pilot signal peripheral circuit simulation with EV. (From bottom to top: (1)PWM from MSP430,(2)Pilot signal,(3)signal at pilot signal monitor)

The simulation shows that it takes around 30 ms to reach the steady state. This value needs to be compensated in the pilot signal monitor firmware.

2.2.2 ZigBee Coordinator

The major function of the ZigBee coordinator is handling the message between the gateway and the end devices. Each time a ZigBee end device joins the mesh network, the coordinator will assign a 16 bits dynamic address to it. In order to pass the command and parameters from the gateway to the desired destination, the ZigBee coordinator needs to recognize and register the unique MAC address of the ZigBee end devices. By using CC2530ZNP with Max3232, the ZigBee coordinator is implemented as shown in Fig. 12, and the firmware flow of our ZigBee coordinator is shown in Fig. 13.

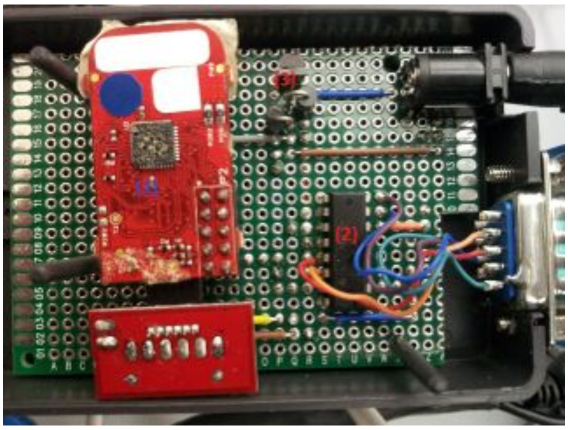

Fig. 12: ZigBee coordinator implementation: (1) ZigBee CC2530ZNP (2) MAX3232

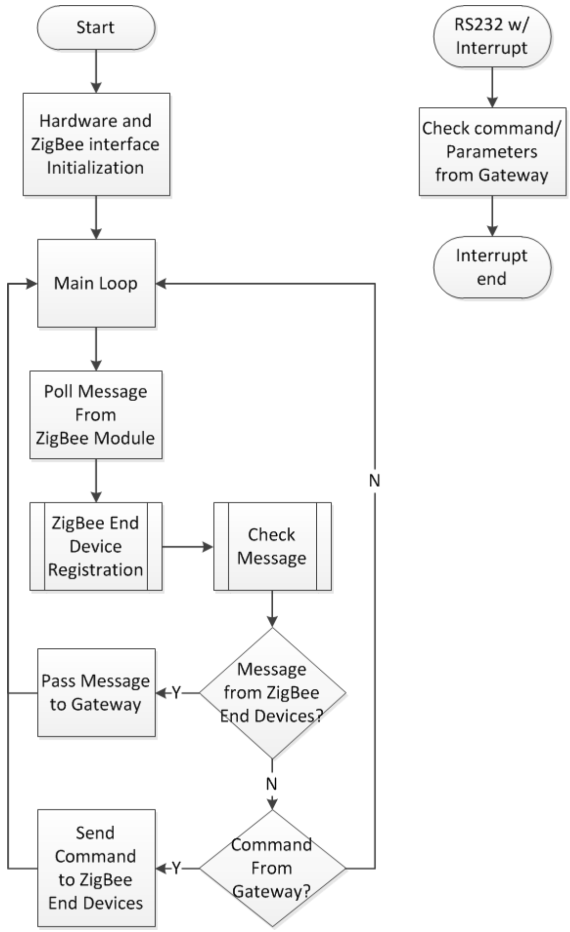

Fig. 13: ZigBee coordinator firmware flow

2.2.3 Server operation

Fig. 14 shows the server's operation flow including enable charging, disable charging and change pilot signal duty cycle during charging.

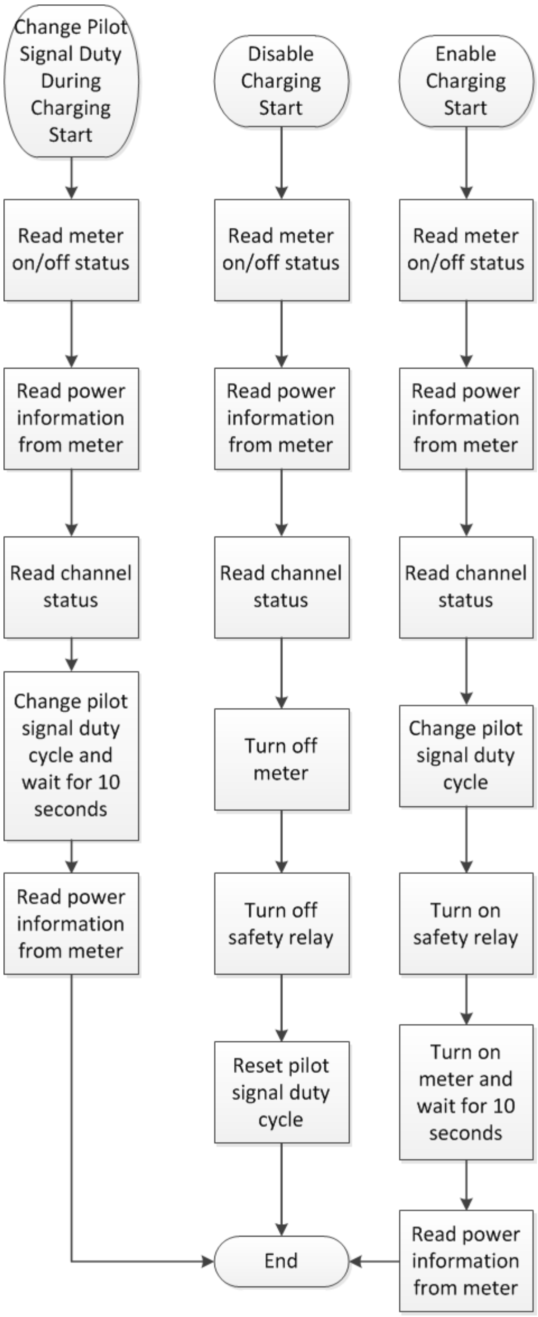

Fig. 14: Server operation flow

In each routine, three processes need to be done including Read meter on/off status, Read meter's power information, and read channel status. The return values of read meter's power information are voltage, current, and active power while the return values of read channel status are pilot signal duty cycle, safety relay on/off status, EV plug in status, and error status. After these three processes, in enable charging routine, the server sends out the command of change pilot signal duty cycle, and then turn on the safety relay and meter. After that, the server will wait for $T_{waiting}$ seconds and send out read power information command. For the routine of disable charging, the server will reset the pilot signal duty cycle after turning off the meter and the safety relay. For the change pilot signal duty cycle during charging routine, the server will change the duty cycle of the pilot signal and read the power information after waiting for 10 seconds. The following format of command is used to send the request to the charging station:

comd [command] [channel] [parameter]

The description of commands and return values of the charging station are summarized in Table 1.

Table 1. Commands and return values of the charging station

Comd	Description and Example

atrs	Auto-reset the whole system, including gateway, meters, relays, and control unit
	comdatrs0000 [return]: N/A
duty	Change the duty cycle
	comdduty0150 [describe]: change channel 1 duty cycle to 50% *duty cycle: 10~85 (6A~51A) [current=duty cycle*0.6] 86~96 (55A~80A), [current=(duty cycle-64)*2.5] [return]: duty0150rely0100plug0101stat0100
enab	Enable EV charging
	comdenab0100 [describe]: enable channel 1 [return]: duty0125rely0100plug0101stat0100
rely	Turn on/off the relay manually
	Ex1: comdrely0101 [describe]: turn on relay at channel 1 [return]: duty0150rely0101plug0101stat0100 Ex2: comdrely0100 [describe]: turn off relay at channel 1 [return]: duty0150rely0100plug0101stat0100
rest	Disable EV charging
	comdrest0100 [describe]: disable channel 1 [return]: duty0100rely0100plug0101stat0100
stat	System statue request
	Ex1: comdstat0100 [describe]: request channel 1 status [return]: duty0150rely0101plug0101stat0100 Ex2: comdstat9900 [describe]: request all channels status [return]: duty0050rely0000plug0000stat0000 duty0150rely0100plug0100stat0100 duty0250rely0200plug0200stat0200 duty0350rely0300plug0300stat0300

3. Experiments and Results

In the experiment, we measure the EV’s response time with the pilot signal change command. The test-bed is a Nissan Leaf with a 110V charging cable. Fig. 15 shows the setup of the experiment.

Fig. 15: Experiment setup of changing pilot signal

To insert and swap our pilot signal with the original signal from the charging cable, a J1772 adaptor is made as shown in Fig. 16.

Fig. 16: J1772 adaptor

Fig. 17 shows the comparison between the original pilot signal (yellow) and our pilot signal (blue).

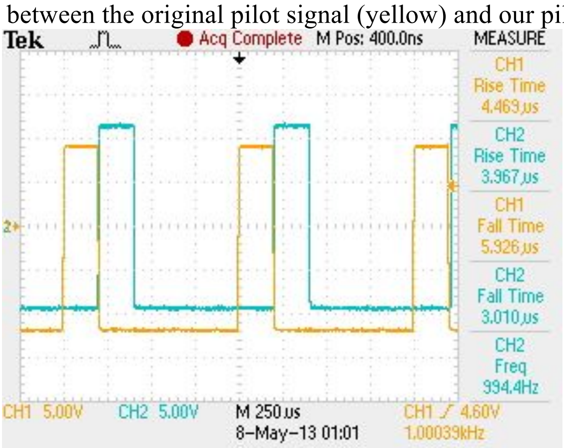

Fig. 17: Comparison between our pilot signal and original pilot signal

The frequency of our pilot signal (994 Hz) is in the range of the J1772 specification (980~1020 Hz). The rise time (3.9 us) and the fall time (3 us) of our pilot signal are slightly over the values specified in J1772 (2 us). However, compared with the pilot signal of commercial charging cable (yellow), our pilot signal (blue) is better in both cases of rise time and fall time, which means our signal would work with commercial EV.

Four cases were tested in the experiments including: (1)0A to 12A, (2)12A to 0A, (3)8A to 12A, and (4) 12A to 8A, which covers the cases of switching between high and low current. The results are show in Fig. 18, Fig. 19, Fig. 20, and Fig. 21. Channel 1 (yellow) is the current taken by the EV while channel 2 (blue) is the pilot signal.

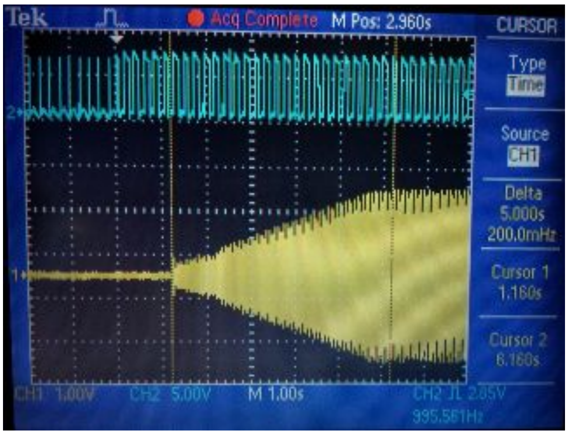

Fig. 18: From 0A to 12A

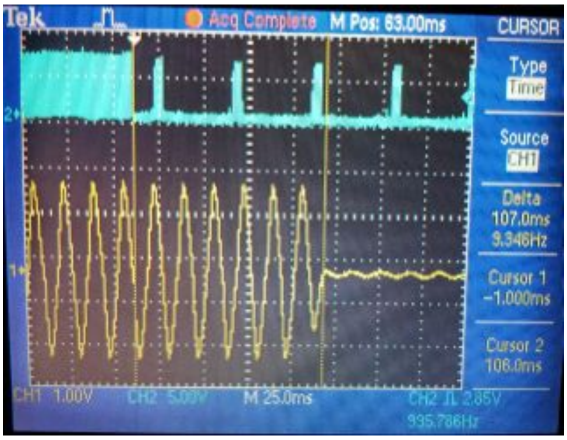

Fig. 19 From 12A to 0A

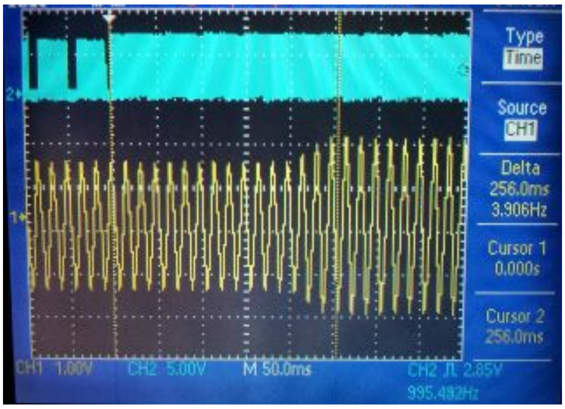

Fig. 20 From 8A to 12A

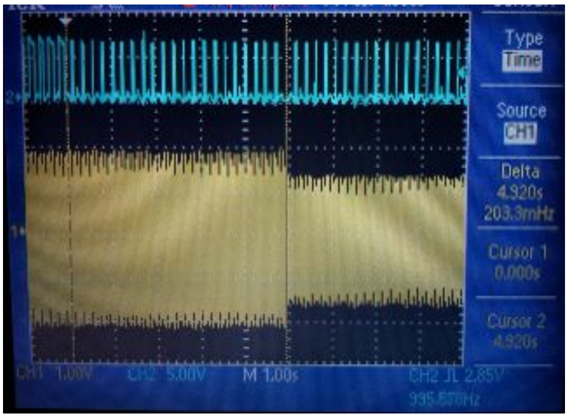

Fig. 21 From 12A to 8A

Note that in the Level 1(110V) charging, even though the server sets the duty cycle larger than 20%, which makes the maximum current capable of being larger than 12A, the Nissan leaf will only take 12A. Here we define T_{EvExe} to be the time between the EV receiving the command and it starting to change the current, T_{EvResp} is to be the time between the EV starting to change the current and when it is settling down. T_{EV} is the summation of T_{EvExe} and T_{EvResp} which can be expressed in (1).

$$T_{EV} = T_{EvExe}(I_{init}, I_{final}) + T_{EvResp}(I_{init}, I_{final}) \tag{1}$$

The experiments are summarized in Table 2. Notice that from the experiments, the T_{EvExe} and T_{EvResp} are related to both the initial current I_{init} and final current I_{final}.

Table 2. EV response time with Nissan Leaf

	I_{init} (A)	I_{final} (A)		
		0	8	12
T_{EvExe} (ms)	0		1000	1000
	8	100		250
	12	100	5000	
T_{EvResp}(ms)	0		3000	5000
	8	10		60
	12	10	60	
T_{EV} (ms) = T_{EvExe} + T_{EvResp}	0		4000	6000
	8	110		310
	12	110	5060	

In the case of I_{final}=0A, T_{EvExe} is 100 ms and T_{EvResp} is 10 ms, which are relatively faster than other cases. It is possible that the EV could turn off its switch in 100 ms without changing its load and 10 ms is the transient response to 0A. In the case of I_{init}=0A, T_{EvExe} is 1000 ms and T_{EvResp} is proportional to the final current. It is possible that the EV turns on its switch in 1000 ms and starts to consume a current proportional to the final current. In the case of I_{init}=12A and I_{final}=8A, T_{EvExe} is 5000 ms. It is possible that the EV's battery management system needs to balance the battery cells and then change its load.

4. Discussion

The timing analysis of the whole system can be further analyzed to improve the performance. The time for returning of changing duty cycle T_{Return} can be expressed as (2):

$$\begin{aligned} T_{Return} &= T_{ControlUnit} + T_{GatewayServer} \\ &= T_{ZigBee} + T_{3gUplink} + T_{Cloud} \end{aligned} \tag{2}$$

Let $T_{waiting}$ be the waiting time after receiving the successful return of pilot signal duty cycle change and before sending the power information request. The waiting time $T_{waiting}$ at server side plug T_{Return} is required to be greater than T_{EV}, which can be expressed in (3).

$$
\begin{aligned}
&T_{Return} + T_{waiting} \\
&\approx \left(T_{ZigBee} + T_{3gUplink} + T_{Cloud}\right) + T_{waiting} \\
&\approx T_{3gUplink} + T_{Cloud} + T_{waiting} \\
&> T_{EV} = T_{EvExe}(I_{init}, I_{final}) + T_{EvResp}(I_{init}, I_{final})
\end{aligned}
\tag{3}
$$

We can rewrite (3) to be (4).

$$
\begin{aligned}
&T_{waiting} > T_{EV} - T_{Return} \\
&\approx T_{EvExe}(I_{init}, I_{final}) + T_{EvResp}(I_{init}, I_{final}) - \left(T_{3gUplink} + T_{Cloud}\right)
\end{aligned}
\tag{4}
$$

From (4), we can see $T_{waiting}$ depends on I_{init}, I_{final}, and the communication traffic. To speed up the system performance, $T_{waiting}$ can be set to be various values depending on I_{init}, I_{final} rather then a fixed value.

5. Conclusion

In this paper, we have proposed, designed and realized a smart charging infrastructure with various current controlling and multiplexing capabilities. Besides the hardware and firmware implementation, the EV's response time to the pilot signal change command is also measured and discussed. In the future, with a smart charging scheduling algorithm implemented on the server, our infrastructure will serve as one of the key components in the national wide smart grid application.

6. Acknowledgement

This work has been sponsored in part by grant from the LADWP/DOE fund 20699 & 20686, Smart Grid Regional Demonstration Project).

7. References

[1] Rajit Gadh et al., "Smart Electric Vehicle (EV) Charging and Grid Integration Apparatus and Methods," PCT International Patent, Ser. No. PCT/US11/40077, June 10, 2011

[2] Rajit Gadh et al., "Intelligent Electric Vehicle Charging System", PCT International Patent, Ser. No. PCT/US12/49393, August 2, 2012.

[3] http://www.coulombtech.com/chargepoint-network.php[03/12/2013]

[4] http://www.blinknetwork.com/network.html[03/12/2013]

[5] http://www.leviton.com/OA_HTML/SectionDisplay.jsp?section=37668&minisite=10251[03/12/2013]

[6] http://www.clippercreek.com/[03/12/2013]

[7] P. Kulshrestha, Lei Wang, Mo-Yuen Chow, S. Lukic, "Intelligent energy management system simulator for PHEVs at municipal parking deck in a smart grid environment," Power & Energy Society General Meeting, PES '09. IEEE, Calgary, Alberta, Canada, July 26-30, 2009

[8] Shun-Neng Yang,Wei-Sheng Cheng, Yu-Ching Hsu, Chai-Hien Gan, Yi-Bing Lin, "Chargescheduling of electricvehicles in highways," Mathematical and Computer Modelling, ELSEVIER, December 6, 2011

[9] Wencong Su, Mo-Yuen Chow, "Computational intelligence-based energy management for a large-scale PHEV/PEV enabled municipal parking deck," Applied Energy, Volume 96, August 2012, Pages 171–182

[10] S. Mal, A. Chattopadhyay, A. Yang, R. Gadh, "Electric vehicle smart charging and vehicle-to-grid operation," International Journal of Parallel, Emergent and Distributed Systems, vol. 27, no. 3. March 2012.

[11] "NIST Framework and Roadmap for Smart Grid Interoperability Standards Release 2.0", Office of the National Coordinator for Smart Grid Interoperability, National Institute of Standards and Technology, U.S. Department of Commerce, February 2012

[12] "SAE Electric Vehicle and Plug in Hybrid Electric Vehicle Conductive Charge Coupler," SAE International Surface Vehicle Recommended Practice. (http://standards.sae.org/j1772_201001/) [03/12/2013]