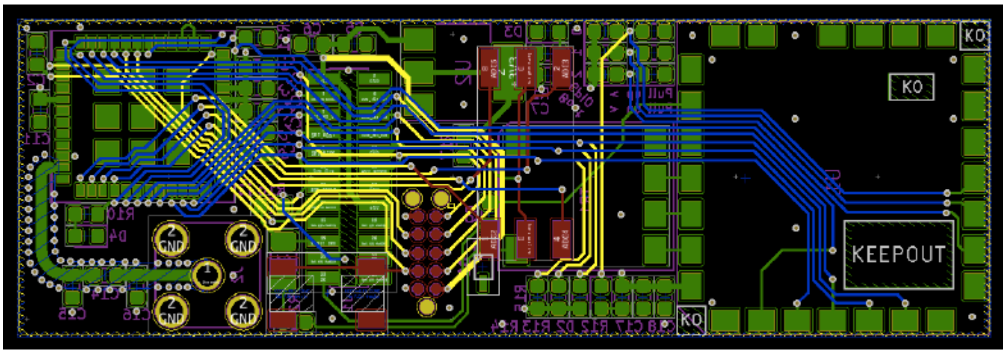
DEGREE PROJECT

Implementation of HomePlug Green Phy standard (ISO15118) into Electric Vehicle Supply Equipment



Rama Pallander

Engineering Physics and Electrical Engineering, bachelor's level
2021

Luleå University of Technology
Department of Computer Science, Electrical and Space Engineering

Abstract

As the use of electric vehicles increases, the need for electric vehicle supply equipment to have more advanced functionality also increases. The HomePlug Green PHY standard was developed to allow more advanced communication between electric vehicles and electric vehicle supply equipment. This more advanced form of communication can solve problems such as load balancing during busy charging and seamless payment methods. There are some modem solutions that are based on the Qualcomm QCA7000 chip that allows for implementation of the HomePlug Green PHY standard.

This thesis explores and highlights the implementation of the hardware for the HomePlug Green PHY standard into a solution that is nearly plug and play for most electric vehicles. A module in the form of a PCB based around one of these modem solutions is developed that allows modular expansion of a traditional electric vehicle supply equipment to gain the functionality of HomePlug Green PHY. The final PCB is a near plug and play solution on the hardware side however, the software needs further development.

Contents

Terms and Designations

EVSE	-	Electric Vehicle Supply Equipment
EV	-	Electric Vehicle
HomePlug GP	-	HomePlug Green Phy
OFDM	-	Orthogonal Frequency Division Multiplexing
ROBO	-	Robust OFDM
BPSK	-	Binary Phase Shift Keying
QAM	-	Quadrature Amplitude Modulation
PLC	-	Power Line Communication
PEV	-	Plug-in Electric Vehicle
CAN	-	Controller Area Network
ECU	-	Elctronic Control Unit

1 Introduction

Ever since the oil crisis of the 1970s, automobile manufacturers have been trying to rid their dependence of crude oil due to its potentially limited quantity. We are now in the dawn of the electric vehicle era which means that not only is this independence possible but we can for the first time since the conception of the automobile reduce its environmental impact to an acceptable level. These new goals however brings new challenges with it, one of such challenge is charging. The main issue with charging currently seems to be speed but that will not be the issue in the near future. Developing faster charging rates is a problem of itself but having a large quantity of automobiles charge at the same time is another issue that has to be solved. Obviously this is not an issue currently due to the low quantity of electric vehicles compared to the overall fleet of vehicles on the roads today. However, even smaller charging stations already today have an issue with charging a hand full of vehicles at their full speed which means they have to contend with evenly splitting the power between all connected vehicles. Imagine in the parking areas of the near future where hundreds, if not thousands of electric vehicles will be connected to the same grid at the same time where this would result in a bad experience for all users. One solution is obviously to throw more power at the problem and while this is tempting this would mean exponential cost increase for the power supplier where expensive fast charging equipment has to be supplied to all parking spots and on top of that the infrastructure and cost required to supply hundreds of cars with more than 100kW of power at the same times would be incredibly costly.

Another solution is to prioritise vehicles on different factors, which include but are not limited to maximum charging speed, maximum charging capacity, current charge level, estimated departure time and estimated distance to next destination. In order to gain this information from the vehicle the local grid has to be able to communicate with the vehicle to extract this information. This is possible due to the already existing HomePlug Green Phy standard which was developed for such applications. The implementation of HomePlug Green Phy also has other use cases, for example it allows for identification of the owner of the vehicle and payment information of said owner which makes the the process of charging and paying a plug and play process. Another use case for HomePlug Green Phy is the possibility of vehicle to grid supply which allows electric vehicle owners to use the battery in their electric vehicle as a backup power supply for their entire home.

At this point it is essential for the reader to understand that the HomePlug Green Phy standard is not developed for electric vehicles specifically. It is designed for any electrical appliance that connects to any piece of copper wire where other devices are also connected. That being said, the HomePlug Green Phy standard is applicable here not only due to what it allows as discussed above but also due to it being very reliable and uses very little energy to do so.

GREPITs Electric Vehicle Supply Equipment is an already J1772 compliant electric vehicle charger. As they are iterating on to this EVSE they decided to create a modular EVSE to use as a platform to evaluate other features, one of these features is HomePlug Green Phy. Since the new iteration of the EVSE is designed from the ground up to be modular, its constituent modules can be designed by different engineers. The purpose of this thesis is to explore and highlight the development of one of these modules. This module will add the HomePlug GP communication capability on top of the J1772 compliant communication. However, due to reasons explained in the thesis, the module will contain additional hardware to be functional so that it can achieve its modularity. This module will let GREPIT AB evaluate how the HomePlug GP implementation will play role in the charging network that they are trying to build and also let them develop the software needed to run on their EVSE platform so they can take full advantage of the HomePlug GP standard.

This thesis will explore and highlight the implementation of the hardware for the HomePlug Green Phy standard into a solution that is nearly plug and play for most EVSE. The reasoning behind this is that if the result is a module then it can not only be used with several different kind of EVSE systems with different form factors but that also means the module can be revised to improve it for future chargers very easily. This thesis starts with a pre-study on what specific hardware and software is required to be compliant with the HomePlug Green Phy standard and if any of it is proprietary. After the correct hardware has been chosen it must then be evaluated to ensure that this implementation is possible. This is done by implementing a development version of the chosen hardware that is designed for such exercises. When compatibility is ensured, the development of the custom hardware or PCB can begin. A custom circuit that is designed around the chosen hardware is then implemented into a PCB that can be added to the rest of the EVSE hardware in the form of a plug in module. When a prototype of the module is produced it is then tested in the same environment and scenario as the evaluation hardware, to ensure that the module and the evaluation kit has the same functionality. Lastly this thesis explores the types of results that can be achieved with the module and discusses the whole process.

1.1 Delimitations

The HomePlug Green Phy standard is very large hence this thesis will focus mostly on hardware and signalling layers of the standard. It is also done within the time frame of approximately 10 weeks where of course the author of this thesis acknowledges allot more could be accomplished with more time.

Furthermore, in order for the software to run on the final hardware developed throughout this thesis, it would need to be ported to a type of hardware it was not written to run on, hence requiring a study of its own.

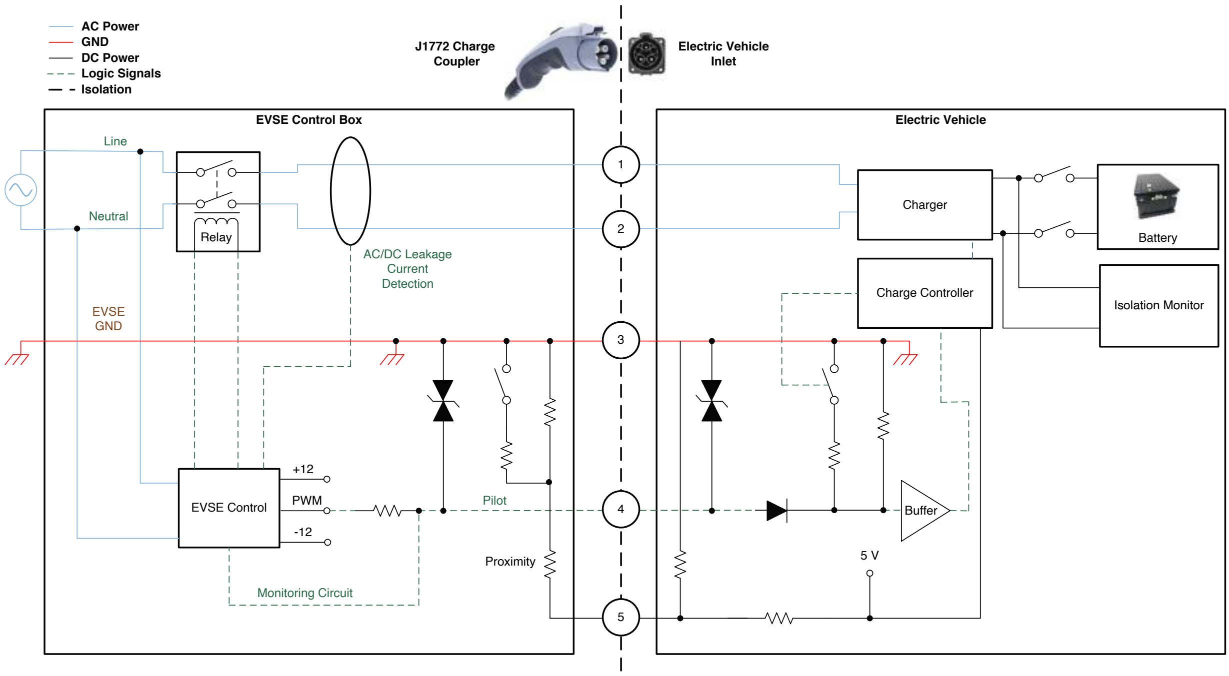


Figure 1: Part of the J1772 hardware near the connector of the EVSE and EV. [9]

2 Theory

2.1 SAE J1772 Standard (IEC 62196)

The SAE J1772 standard is the most common standard used in Electric Vehicles currently, this includes the prototype EVSE. It covers aspects such as physical requirements, electrical signaling, communication protocol, connectros, and performance requirements for the electrical vehicle conductive charging system. The aspects of the standard that are relevant are discussed below. The reader should be aware that most of the information highlighted in this part of the thesis is from one source, therefore in order to avoid repeated citations, the reader should be aware that unless specified, the information is gathered from source [9].

2.1.1 The connector and the pinout

The connector that can be seen in Figure 2 was first introduced in the J1772-2009 standard and has 5 conductors. It is more commonly known as the "Type 1" connector, and the conductors include,

- AC Phase 1,
- AC Neutral,
- Control Pilot (CP),
- Proximity Pilot (PP),
- Ground.

Later there was a connector introduced that has two additional pins which adds two more AC phases known as the "Type 2" connector. The addition of two more AC phases allow for higher power transfer to the EV thanks to a three phase system. Additionally the PP signal is used for the EVSE and EV to detect each other so and is used as a safety measure

Figure 2: Image of the J1772 compliant connector known as "Type 1"

to prevent the user from driving away while the EV is still plugged into the EVSE. It is also directly linked to the latch of the connector.

2.1.2 The Control Pilot Signal

The Control Pilot signal is where the PEV signals to the EVSE its status and readiness. The CP is a ±12V PWM square wave that is altered by the PEV to indicate different statuses, however, the CP signal is generated by the EVSE. The square wave always has a negative cycle that goes to -12V but the positive half of the cycle is manipulated which is what indicates different states, and that is done by switching in different resistances and diodes as seen in Figure 1. A tabled explanation of the different states is shown in Table 1, where also the different resistances as presented to the the EVSE can be seen.

State	Pilot High voltage (V)	Pilot Low Voltage (V)	Frequency (kHz)	Resistance (Ω)	Description
State A	+12	N/A	DC	N/A	No connection
State B	+9	-12	1	2740	EV connected, Ready to charge
State C	+6	-12	1	882	EV charging
State D	+3	-12	1	246	EV charging, needs ventilation
State E	0	0	N/A	-	Error or no power
State F	N/A	-12	DC	-	Unknown error

Table 1: The meaning of the different states of CP signal

The duty cycle of the PWM signal is controlled by the EVSE and tells the vehicle the amount of current that is available to the vehicle. The duty cycle that is to be selected is controlled by two equations. There is one equation for systems which can deliver between 5 amps and

51 amps and that is

$$Dutycucle = \frac{Amps}{0.6} \quad (1)$$

and another equation for system which can deliver 51 to 80 amps, which is

$$Dutycucle = \frac{Amps}{2.5} + 64. \quad (2)$$

Note that the highest duty cycle that equation 1 can give is 85% and the lowest duty cycle that equation 2 can give is 84.4% which means they overlap nicely. However the lowest duty cycle that equation 1 can give is 8.33%. So when the EV is presented with a PWM signal with a duty cycle of 5% or lower then it goes into Digital communication mode and communicates with the EVSE with that protocol. The definition of that protocol is covered by the HomePlug Green Phy Standard.

2.2 HomePlug AV Standard

The HomePlug Green Phy standard is a simplified version of of the HomePlug AV standard. Hence the HomePlug AV standard is outlined first in this section. This will be a very simplified version of the HomePlug AV standard in order to keep this thesis concise. However a slightly less shortened version of the standard, in the form of a white paper, is cited in the references and the reader can read more in depth information there. Since there is allot of information from said white paper in this section, in order to avoid repeating the same source too much, it can be assumed by the reader that the information can be read from the white paper unless otherwise specified [8]. The reason that HomePlug GP is a simplified version of HomePlug AV is that it does not need as high speeds and the fact that simplification provides more robustness this case.

2.2.1 Overview of the HomePlug AV physical Layer

The basis of the physical layer of HomePlug AV can be outlined in a simple table as seen in Table 2.

Spectrum	2 MHz to 30 MHz
Modulation	Orthogonal Frequency Division Multiplexing
Number of Subcarriers	1155
Subcarrier Spacing	24.414 kHz
Subcarrier Modulation Formats	BPSK, QPSK, 16 QAM to 1024 QAM
Data FEC	Turbo code: Rate 1/2 or Rate 16/21 (punctured)
Supported data rates	ROBO: 4 Mbps to 10 Mbps Adaptive Bit Loading: 20 Mbps to 200 Mbps

Table 2: Key Attributes of HomePlug AV PHY layer

Orthogonal Frequency Division Multiplexing (OFDM) means that information is distributed among equally spaced narrow band subcarriers instead of one single broadband carrier. At a signal bandwidth between 2MHz and 30MHz the HomePlug AV signals will be identified as high frequency noise by most electronic devices that do not use the HomePlug AV standard. In addition, due to the frequencies being so high, most devices are already equipped with input filer capacitors that attenuates the signal and hence the HomePlug AV modulation will not disturb appliances that do not use it.

2.2.2 Adaptive Bit Loading and Tone Maps

Any broadband signal transmitted over the home power lines will have issues with multipath distortion. Multipath distortion is when a signal reaches the receiver multiple times over multiple paths. When the signal recombines at the receiver it can encounter different levels of attenuation due to destructive interference. To combat this the HomePlug AV standard was outlined with OFDM capabilities. This provides excellent robustness against frequency selective fading.

HomePlug AV also supports adaptive Bit Loading. This means that each subcarrier can be adapted to have a certain bit depth. A very attenuated subcarrier can use Binary Phase Shift Keying (BPSK) which will give it a bit depth of 2 bits, while a very low attenuated subcarrier can support 1024 Quadrature Amplitude Modulation (QAM) giving it a bit depth of 10 bits. This allows each subcarrier to carry as much data as the conditions will allow.

In order for all this to be possible, the transmitter has to know the attenuation level of each subcarrier at the receiver. Thus HomePlug AV devices on the same network will periodically exchange sounding packets and build tone maps to keep track of the attenuation levels of each subcarrier for all receivers on the network. All HomePlug AV devices are transmitters and receivers, hence this results in all devices keeping track of tone maps for all receivers on the network. This means that with a network of N devices, each device has to store $N - 1$ tone maps. While this does introduce a high degree of complexity to the network, it allows speeds of up 200 Mbs to be reached.

2.2.3 Central Coordinator

Each HomePlug network requires a Central Coordinator (CCo). Any device on the network can act as CCo and if its removed from the network then any other device on the network can fill the CCo role. The best device suited for the CCo job is selected by the network. This selection occurs based on a set of pre-defined selection criteria. The CCo periodically transmits a beacon that the rest of the network synchronises to. This period is synced to the AC line frequency.

2.2.4 Other attributes

As explained earlier, the HomePlug AV is a long and complicated standard and all of it cannot be outlined completely in this thesis. However, there are a few more aspects of the standard that the reader should be aware of but do not need an in depth discussion.

- **Carrier Sense Multiple Access** - HomePlug AV uses Carrier Sense Multiple Access (CSMA) as the basic channel access priority mechanism. There are also three other priority levels, CAP1, CAP2, and CAP3. Since this is identical between HomePlug AV and GP and does not affect the thesis, it will not be discussed further
- **Time Domain Multiple Access** - As an option, HomePlug AV supports Time Domain Multiple Access (TDMA). Since this feature is completely omitted from HomePlug

GP then it is thus not relevant to this thesis.

2.3 HomePlug Green Phy Standard

As stated before, HomePlug GP is a derivation of HomePlug AV and hence it is fully interoperable with HomePlug AV and its constituents.

Parameter	**HomePlug AV**	**HomePlug GP**
Spectrum	2 MHz to 30 MHz	2 MHz to 30 MHz
Modulation	OFDM	OFDM
Number of Subcarriers	1155	1155
Subcarrier Spacing	24.414 kHz	24.414 kHz
Subcarrier Modulation Formats	BPSK, QPSK, 16 QAM to 1024 QAM	QPSK
Data FEC	Turbo code: Rate 1/2 or Rate 16/21 (punctured)	Turbo code: Rate 1/2
Supported data rates	ROBO: 4 Mbps to 10 Mbps Adaptive Bit Loading: 20 Mbps to 200 Mbps	ROBO: 4 Mbps to 10 Mbps

Table 3: Key Attributes of HomePlug Green PHY layer.

2.3.1 Physical Layer

As seen in table 3, HomePlug GP is a simplified version of HomePlug AV. These simplifications have a reason and these reasons are detailed below.

HomePlug GP uses the same 2 MHz to 30 MHz frequency band, the same modulation and the same number of subcarriers. The forward error correction is also the same with turbo codes. However the HomePlug GP standard emphasises reliability over speed so the data throughput was limited to 10 Mbps as opposed to 200 Mbps which also resulted in lower power consumption. This limitation is the result of two key factors of HomePlug GP.

1. Restrict the OFDM subcarrier mode to QPSK exclusively
2. Only use ROBO modes which eliminates the need for adaptive bit loading

Although this restriction might seem harsh, the HomePlug Alliance deemed this speed sufficient not only because it saves costs but it also future proofs the HomePlug GP standard since it is still 1000 times faster than competing Power Line Communication (PLC) solutions such as Prime or G3.

2.3.2 ROBO Modes

Robust OFDM (ROBO) is a form transmission where the information is sent redundantly on multiple carriers. This gives HomePlug GP a much higher degree of reliability. In fact there are several ROBO modes supported by HomePlug GP where it can intelligently decide whether to priorise speed or even more robustness.

The use of ROBO also eliminates the need for shared tone maps which reduces complexity immensely. Given that most smart grid applications require less than 250kbps the speeds at which ROBO limits to are more than adequate. As a whole the system has a headroom but is still very reliable.

Mode	PHY Rate	Number of repeats
Mini-Robo	3.8Mbps	5
Standard Robo	4.9Mbps	4
High Speed Robo	9.8Mbps	2

Table 4: ROBO modes supported by HomePlug GP and the speed trade off.

2.3.3 MAC Layer and other features

The MAC layer of HomePlug GP is very similar to that of HomePlug AV, However there are some small differences. HomePlug GP uses the same CSMA access priority system but without the optional TDMA. In addition there are HomePlug GP exclusive features to ensure interoperability with HomePlug AV. A power saving mechanism was added to reduce power consumption even more and a bandwidth sharing algorithm was also developed. These two features can be explored in more detail in the white paper but shall not be discussed here.

Lastly a method for associating and binding with plug-in electric vehicles on public charging networks was added. This mechanism works by characterising signal level attenuation of the vehicle. While this sounds similar to the bit loading tone maps discussed previously, it is not the same.

2.3.4 Plug-In Vehicle Association

As the numbers of EVSE locations grow combined with the fact that billing is essential for this equipment to function, the necessity for secure billing and correct vehicle association will grow. EVSE will be present in locations such as malls, convenience stores, dentist offices, and among other places. Thus for the majority of time, vehicles will be operating withing close proximity of each other. This could vary from two PEV's connected to a two channel EV charger to dozens of PEV's connected to the same grid in a small bussiness parking lot. It is inevitable that the signals from one PEV will be heard by multiple EVSE or vice versa.

To combat this, a feature was added to HomePlug GP called Signal Level Attenuation Characterisation (SLAC). As part of the "Mathing EV - EVSE process" [4], the SLAC process works by the PEV sending a number of SOUNDING packets and each EVSE withing hearing range calculates the received average signal power and sends back the information to the same MAC adress from which the PEV sent the packets . The vehicle then matches with the EVSE that has the highest signal level. The PEV gets the MAC adress of the EVSE when the EVSE sends the calculated result back to the PEV.

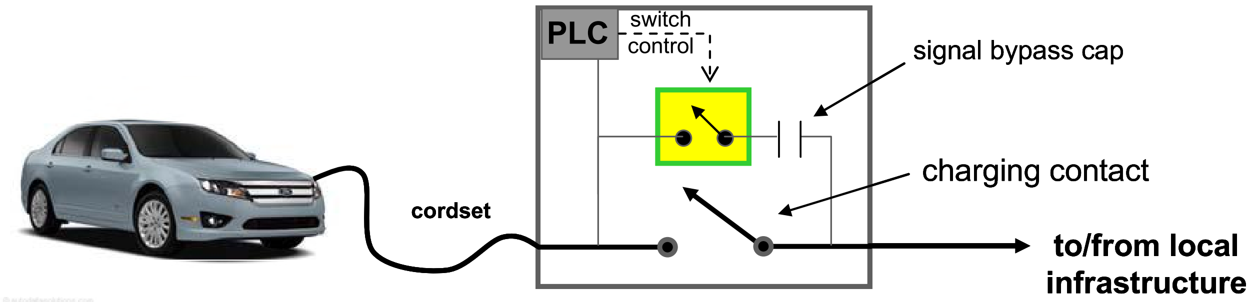


Figure 3: A PEV on the left connected to an EVSE. The box on the right shows a simplified version of a small part of the EVSE circuit. Both signal switch and charging contact switch are open by default. Source: [8].

When the PEV and EVSE first connects, the contacts for the main charging and the PLC signal are open. When the PEV is plugged in it starts sending PLC signals. The combination of the switches being open and that capacitor in series attempts to prevent the signals from propagating past tehe EVSE and onto the infrastructure grid. The PLC signal can still propagate past the EVSE but only with large amounts of attenuation. There are other ways that the PLC signal can couple onto other cables but again only with a high degree of signal power loss. Therefore the PEV PLC signal will always have the highest power at the nearest EVSE. This ensures in combination with the SLAC process, EVSE/PEV association is reliable and secure.

3 Method

3.1 Reasoning behind hardware chosen

So far the HomePlug Green Phy standards communication signals have been discussed theoretically. In order to make them into reality, the simplest way is to do it through a modem that has been specifically made to the standards outline. A modem will allow a computer to send and receive HomePlug GP signals over any copper in a complete package. There are many types of Power Line Communication (PLC) modems on the market but there seems to be only one HomePlug Green Phy specific modem and that is the Qualcomm QCA7000.

Figure 4: Image of the PLC Stamp Micro 2 [7]

For this prototype EVSE, the choice is to work with a module supplier. While there are a few of these suppliers on the market, one of the main features of the prototypes EVSE circuitry is that it can fit in a traditional block heater locker casing meaning there are size restrictions. Due to these restrictions, there is one product that the *HomePlug GP module* could be based on that worked exceptionally, this is the *PLC stamp micro 2* by I^2SE. It should be clarified that between the time of developing the module and the time of writing this thesis, I^2SE seems to be under new management and have changed name to *intech*. However, their products seems to have stayed the same. There are some small advantages to using a ready made implementation of a chip. When the chip implementation is developed, a lot of the brunt work is already done. This includes but is not limited to:

- Power filtering for the QCA7000 chip is already done on the module textitstamp micro.
- SPI and UART communication interfaces are ready and simpler to connect to with a host computer or controller chip.
- A memory IC is included to store the firmware which also simplifies flashing of said firmware.

3.2 Integration

As stated in the introduction, this prototype that is being developed is a J1772 and HomePlug GP EVSE solution so how will the *HomePlug GP module* integrate with it? The goal of the

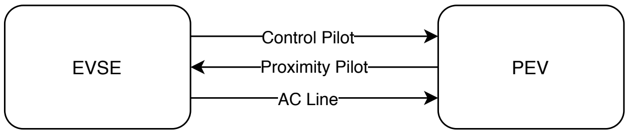


Figure 5: How a J1772 compliant EVSE connects with a Plug-In Electric Vehicle (PEV).

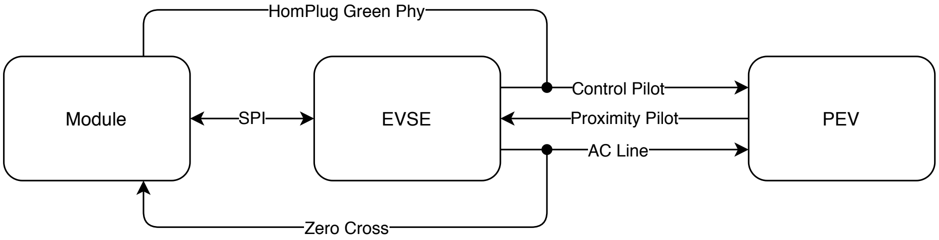


Figure 6: System that is compliant with HomePlug Green PHY.

new prototype EVSE was to be modular so that some functionality could be easily removed or added without affecting the other aspects of the EVSE. Hence the HomePlug GP standard is implemented as an add-on module. A part of a J1772 compliant system is shown in Figure 1 and a simplified version of a whole J1772 compliant EVSE is shown in Figure 5.

As HomePlug GP uses a PLC modulation, it can modulate over 110V AC or 230V AC lines however, there is also the option of modulating over the Control Pilot wire. Since the Amplitude of the HomePlug GP signal is so low and the frequency so high compared to the Control Pilot signal, there is no interference with the older standard. This results in a solution where the *HomePlug GP module* couples onto the EVSE as shown in Figure 6.

The zero cross signal that is generated from the AC line is used by the module to time sync the messages. SPI is a common communication protocol used by many different devices, if the reader wishes to learn more about SPI, more information can be found in [3].

3.3 Evaluation

3.3.1 Hardware

In order to make sure that the *PLC stamp micro 2* will work as intended with the prototype EVSE and a real world PEV it needs to be evaluated. I^2SE of course also offer an evaluation board (EVB) for the *PLC stamp micro 2*. This board offers all the connectivity that the *PLC stamp micro 2* is capable of and makes them available simultaneously in a break out board solution. The EVB also has a USB connector giving the user the option of powering the board with any USB power source or any of the power pins on the pin cluster that is located on the north end of the board as seen in Figure 7.

In Figure 8 we can see the different inputs and outputs offered by the *stamp micro* and EVB in combination. According to Figure 8 it is evident that the *stamp micro* can operate over the AC lines or the Control Pilot wire. This module was chosen to use the Control Pilot wire to transmit data so the 1:1:1 transformer on the EVB is used. The 5V supply voltage was going to be supplied from an external source, the EVSE main PCB in which this HomePlug GP module is going to be connected to. The EVB is going to be directly connect to the AC lines of the charger because the EVB already has a circuit that generates a zero cross signal that the *stamp micro* can interpret. Lastly, the SPI connectors are going to be connected to a Raspberry Pi that can control the *stamp micro*. This is needed simply because the

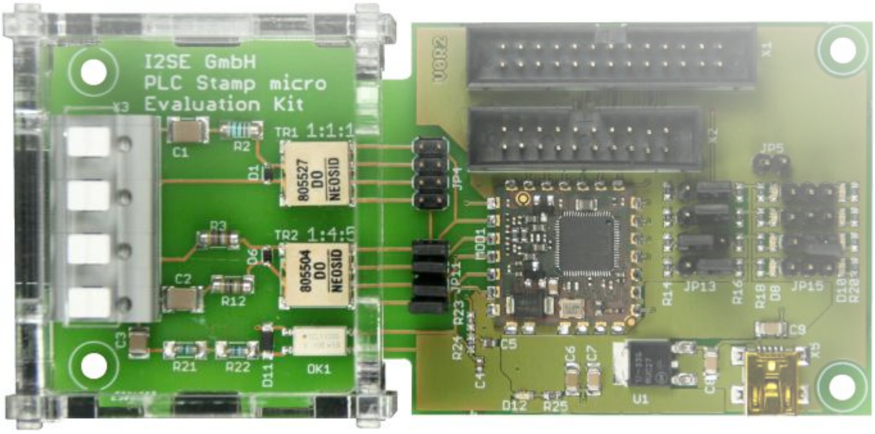


Figure 7: Picture of the evaluation board from the top. The *stamp micro* can also be seen. [6]

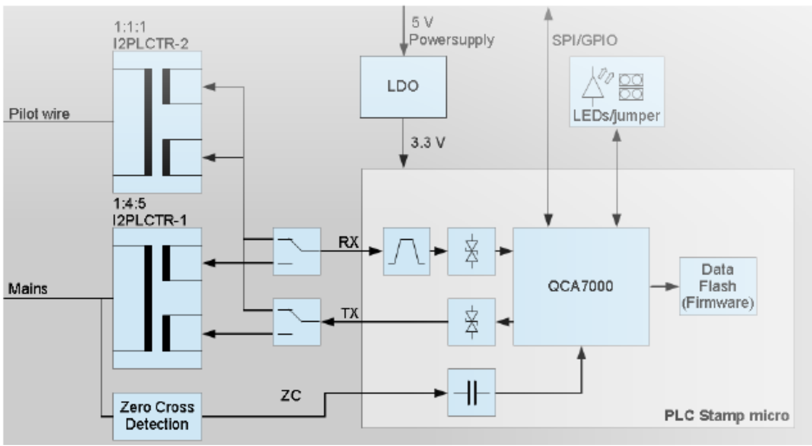


Figure 8: The diagram layout of the EVB [6]

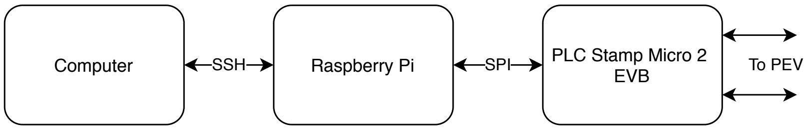


Figure 9: Basic layout of the evaluation chain.

stamp micro is only a modem that encodes and decodes HomePlug GP signals or packets. This means that there still needs to be a computer or some kind of logic that interprets the information within the packets and can act upon that information. Hence for the sake of the evaluation, the single board micro controller known as a Raspberry Pi takes on these duties.

The whole chain of evaluation hardware can be seen in Figure 9. The *stamp micro* encodes and decodes HomePlug GP packets and then reports it to the Raspberry Pi which in turn interprets the information and responds accordingly. The computer that connects to the Raspberry Pi is there to start the appropriate software, to read debugging messages and to monitor the status.

Finally, the EVB has four GPIO pins. These pins have two functions. During boot-up they tell the *stamp micro* what configuration to go into. Post boot-up they act as status LEDs for various functions and the LEDs can be chosen to be low active or high active. Low active means the LED is on when that signal is low and high active means the LED is on when the signal is high.

3.3.2 Software [2]

The reader should keep in mind that the *stamp micro* is a simple modem. This means that all the correct order of packages and timings and such things that are outlined by the HomePlug GP standard is controlled by software. Thankfully there is an open source software package made by a small community ready that was programmed to follow the standard. It is written in C and has its own C-based compiler that outputs files that run without a hitch on the raspberry pi systems that it was tested on during this development cycle. The whole package consists of 2959 files including the documentation and is too convoluted to outline in this thesis. The basic functionality of the code is that it takes the necessary data, packages it and sends them to the *stamp micro* over SPI which then sends it over the copper as ethernet packages. The received data from the PEV is then decoded by the *stamp micro* and then over SPI, the *stamp micro* reports back to the Raspberry Pi on which the software that is running interprets the data. There is only one issue about this software package, it was last updated in 2014 at the time of writing and since then the HomePlug GP standard has had changes made to it. The consequences of this will be evident later in the thesis.

3.4 Development of the HomePlug GP module using the *stamp micro*

During development the logical setup of the hardware will not look as it does in Figure 6. However, the the setup in Figure 6 is the end goal. Two *stamp micros* were ordered since they come in two different configurations. There is one PEV variant and an one EVSE variant. This is to cover as many test scenarios as possible.

Development process for this module was done in such a way that the first setup was to be completely ideal where we can ensure that everything works as intended. With each subsequent step, one part of the chain is swapped and the functionality is tested. If everything still works as intended then this step is repeated until the development arrive at the scenario seen in Figure 6.

3.4.1 Desk scenario

The software that runs on the Raspberry Pi can either emulate the EVSE side packets or the PEV side packets. This means that we can connect two EVB boards and their respective Raspberry Pi's together to observe the communication in a real world test.

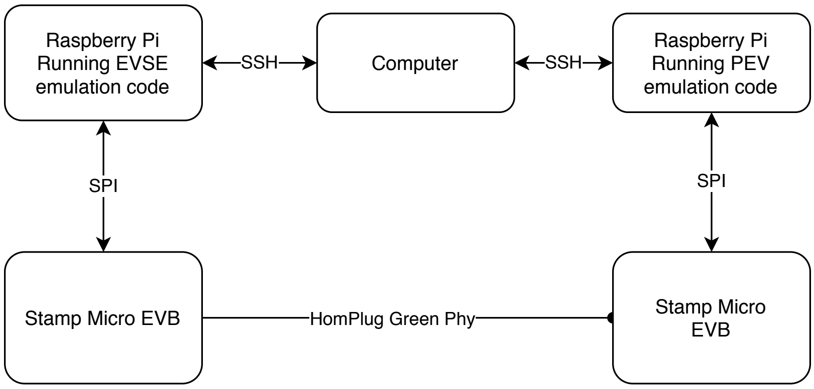


Figure 10: Two EVB boards with *stamp micros* in each talking to each other.

In Figure 10 the most basic HomePlug GP network running can be seen. This is the starting point of the development process. Each step will later replace a portion of the setup to get close to a real world scenario as seen in Figure 6. During testing when this configuration was up and running the following occurs:

- Matching process begins
- Matching Process succeeds
  - Matching process is outlined in appendix B in figures A.1 and A.2 where an extract of [4] can be found.
- Charging starts
  - It should be reiterated that this is a desk scenario with just two EVBs connected to each other with a Raspberry Pi. In a real world scenario this would mean that the EVSE switches the relays for the AC line.
- Charging ends after predetermined time
- Connection is closed
- EVSE starts listening for next connected PEV

This is a desired output. This means that every step of the process works as intended.

3.4.2 Add a real PEV

The next step is to replace the emulated PEV with a real PEV. If the EVB emulating an EVSE connects with a real PEV through the cabling of an already existing EVSE successfully then this means that the module implementation is also successful. In this configuration, a

J1772 type 2 cable was modified so that the EVB could be connected as shown in Figure 11. A J1772 compliant EVSE was modified to tell the PEV to go into a mode called *digital communications mode*. This simply tells the PEV that charging information, such as maximum number of amps available will not be communicated over the traditional J1772 control pilot signal but this information will be given through HomePlug GP packages. The PEV goes into *digital communications mode* when the control pilot PWM duty cycle is set to 5% [5][4]. As discussed previously in the theory section, the PWM duty cycle tells the PEV the maximum amps available for charging and the table starts at 8.33%. To get the EVSE to do this, it was flashed with a custom firmware that had a control pilot duty cycle locked to 5% at all times and this was never changed. In this scenario, the following occured.

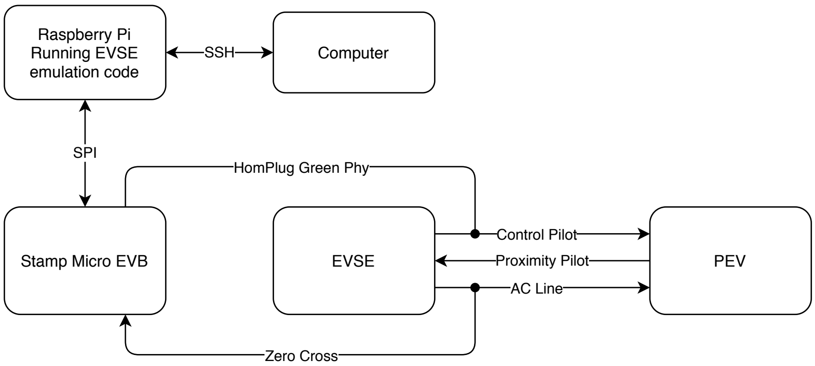


Figure 11: EVB connected to a real PEV

- Matching process begins
- Matching Process fails
- Connection is closed
- EVSE starts listening for next connected PEV

Due to the Matching process failing, this is not a desired outcome.

This was a great block in the development process since the reason behind the issue was not clear. When changing from an emulated PEV to a real world PEV, both the hardware and the software was swapped out on the PEV side. Since the PEV is a real vehicle that is in production, it is assumed that bugs are extremely rare and that the implementation is done correctly. This means that the issue was on the EVSE side, After consultation from I^2SE, they concluded that the hardware setup on the EVSE side was done correctly so that meant the issue was the software running on the Raspberry Pi. This was not a great surprise since the code had not been updated since 2014. After days of debugging it was revealed that during the matching process, the code did not send the correct MAC address to the PEV since it was expecting the PEV to just read the MAC address from the Ethernet package header. To clarify even further, in the outlined matching process that can be seen in the appendix the fault was specifically with the *CM_SLAC_PARM.CNF* package. In the previous step when the PEV sends the *CM_SLAC_PARM.REQ* package, the EVSE software is supposed to read the MAC address of from the header of the *CM_SLAC_PARM.REQ* package header and insert it not only in the *CM_SLAC_PARM.CNF* package header but also in the package at Octet 9 to 14. This requirement can be seen in A.2 following figures A.1 and A.2 in appendix B. After the appropriate lines of code had been changed the full process ran perfectly. Hence, the following occurred

- Matching process begins

- Matching Process succeeds
- Charging starts
- Charging ends after predetermined time
- Connection is closed
- EVSE starts listening for next connected PEV

At this point, the EVB had served its purpose and the concept was proven to work. This means we knew that if we were to implement the *stamp micro* into its dedicated PCB and connect it to the EVSE it will give the EVSE the ability to communicate as HomePlug GP device.

3.5 Designing the circuit

All the necessary components needed to run the *stamp micro* properly is outlined in its data sheet [7] which means that the [7] data sheet will be used as a reference to design the circuit around *stamp micro*. This brings up a new issue, for obvious reasons, a raspberry pi will not fit into the EVSE case. As previously stated in this thesis, the *stamp micro* is simply a modem and hence needs another device to tell it what data to send and to interpret the data that it receives. However, the prototype EVSE already has a component that can take on such duties. It is another electronics module that is used to send receive data over the air and contains a *STM32L* micro controller [10]. However, despite it being a module, to avoid confusion it will be referred to as the *Radio IC* from now on. In this implementation, not only will the *Radio IC* do its normal job of communicating over the air but also tell the *stamp micro* what to do and act upon the data that the *stamp micro* receives. The configuration can now be seen in Figure 12 which is as close as we can get to our goal in Figure 6. This means that the *Radio IC* and *stamp micro* will be on the same PCB module in the prototype EVSE.

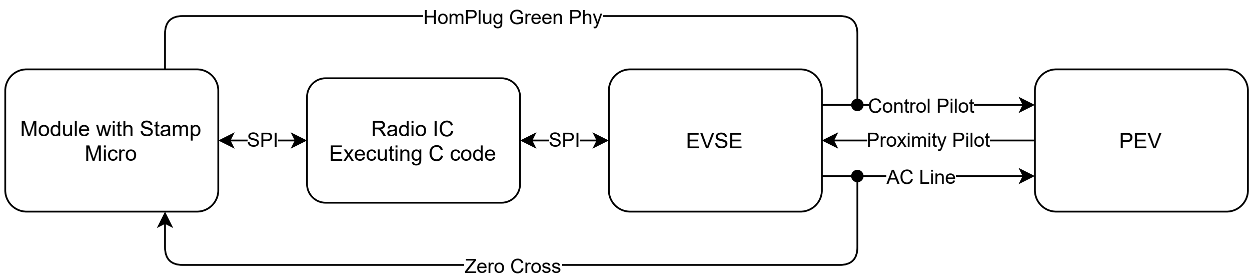


Figure 12: Final configuration of the *HomePlug GP module*

3.5.1 Schematic

The EVB had a lot of pins, especially on the north side of the PCB that can be seen in Figure 7, however, the *stamp micro* simply does not have that many pins as can be seen on the data sheet. Only a few pins on the north side connector are used. Lets begin by looking at the *stamp micro* itself.

A seen in Figure 13 a lot of the pins are just ground pins but most of the other pins are being used in this implementation. All of these subsequent pins that are used will be explained but the reader should direct their attention to the zero cross signal coming from an external

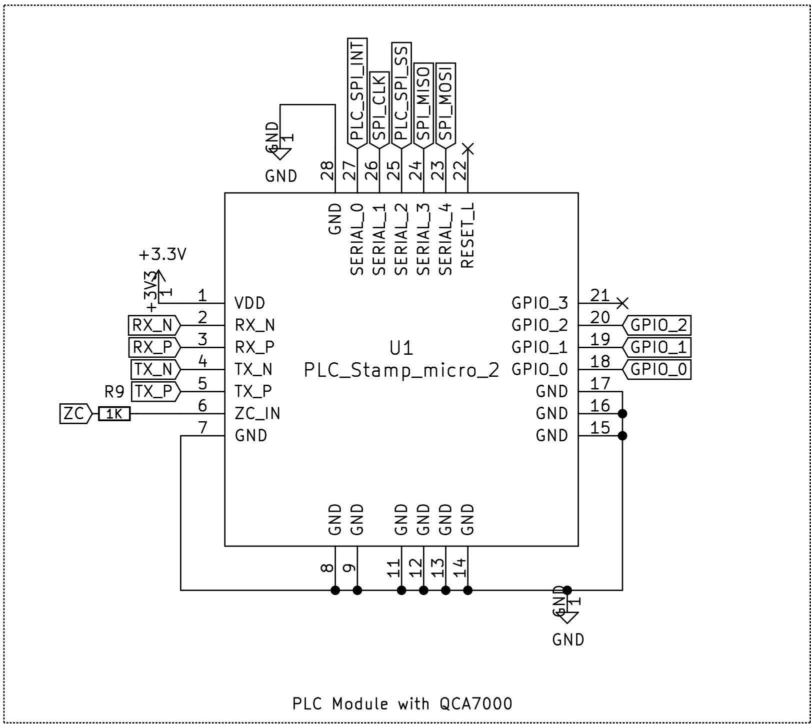


Figure 13: Circuit showing the connections that the *stamp micro* has available to it.

part of the PCB directly into the *stamp micro* through a termination resistor. The EVB for the *stamp micro* had a opto-transistor in a circuit that generated a zero cross signal in the shape of a triangular wave which then was fed into the *stamp micro*. This is not needed in this case since the EVSE main PCB already provides a zero cross signal from elsewhere which is then directly sent to the *stamp micro*.

In Figure 14 we can see the coupling transformer used for the *stamp micro* so that it can couple its signals onto the Control Pilot wire. The capacitor on the input is to block any DC offset from the signal which can occur. The TVS diode is there for safety where it helps to protect any other semiconductors from large voltage transients. The observant reader might have also noted that the RX negative and positive signals have been flipped, and the same is the case for the TX signals as compared to the data sheet [7]. This is due to that the

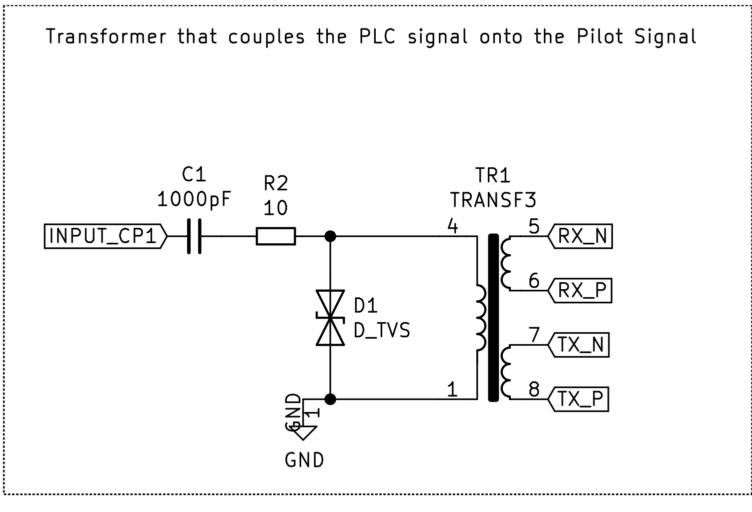


Figure 14: Circuit for the *stamp micro* to couple onto the Control Pilot wire

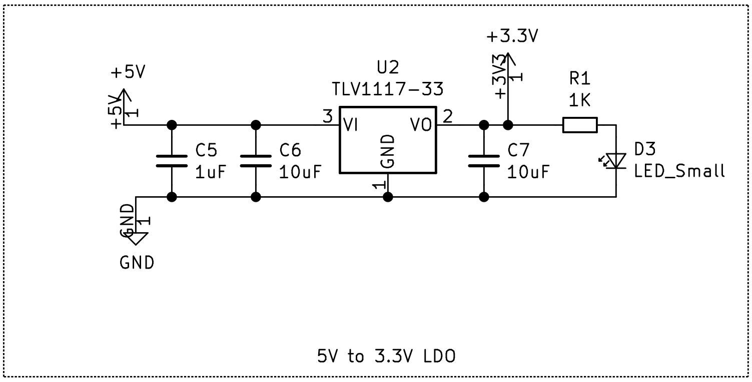


Figure 15: Circuit for the Low Dropout (voltage) Regulator, also known as an LDO

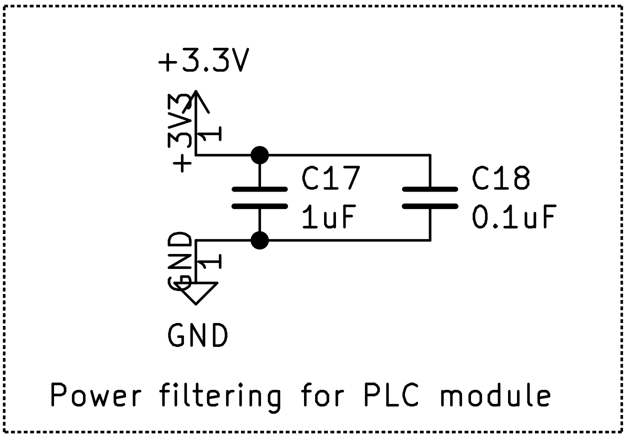


Figure 16: Power filter for the *stamp micro*

transformer Figure in the data sheet depicts a non inverting transformer while the real world transformer used is an inverting transformer. Furthermore, in the *stamp micro* data sheet, the positive and negative part of both RX and TX can be seen to cross one another and then connect to the stamp micro. This however results in messy routing on the PCB. So since the transformer used is inverting and the positive and negative part of the signals is flipped, they don't have to cross each other. This means that the signal on both secondaries will be flipped already so they don't have to cross each other and hence, resulting in a cleaner routes on the PCB.

Since the *stamp micro* needs $3.3V$ [7] to run and the only external voltage source is $5V$ a Low Dropout Regulator (LDO) is needed to step down the voltage. In Figure 15 we can see such circuit. The LDO also helps with stabilising any ripples or variations in the voltage. The capacitors on the input and the output are there to stabilise the voltage even further. The LED in the circuit is there as a diagnostics tool to quickly determine if the board is getting power or not.

Two more capacitors are placed close to the input power port of the *stamp micro* to ensure that any noise picked up by the $3.3V$ rail is filtered out. These capacitors can be seen in Figure 16 and have different values. The capacitor with the higher capacitance helps reduce lower frequency noise with higher energies and the capacitor with the lower rating helps reduce higher frequency noise that has lower energy levels.

Finally the GPIO pins can be seen in Figure 17. This circuit might look odd to the reader

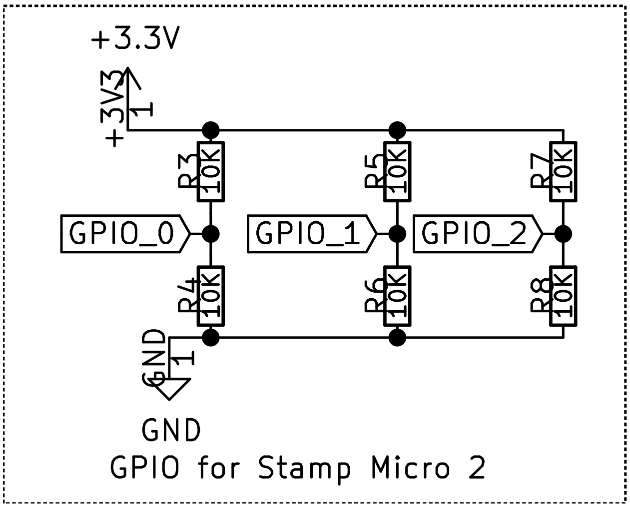


Figure 17: GPIO Pins have pull down and pull up resistors but both are not populated at the same time.

and at first glance it should, since the GPIO pins require a pull up resistor or a pull down resistor. Depending on whether the resistor is pull up or pull down the *stamp micro* reads these pins on startup where pull up activates one function and pull down activates another. For example $GPIO0$ determines whether the *stamp micro* should boot up with the firmware given by the host IC or whether it should boot up with the firmware it can read on the flash memory. Having a pull up resistor and a pull down resistor present in the circuit means that both footprints will be available on the PCB and by populating and repopulating the relevant resistors, the user can choose what configuration to run the *stamp micro* in.

3.5.2 Choice of components and footprints

The component size chosen needs to find a good balance between space saving and hand solderability. As outline further down, there is a size constraint for the module. Hand solderability is useful for repair purposes when a specific component needs to be swapped out.

The resistors and capacitors chosen are a 0603 size, which means that they are roughly 0.06 inches in length and roughly 0.03 inches in width. This size is a good compromise between hand solder ability and space saving. However the input capacitor for the Control Pilot signal that can be seen in Figure 14 is a larger 1206 footprint since it is necessary for it to handle a larger power load. The *stamp micro* was chosen for its small footprint from the beginning. The coupling transformer and TVS diode were copied directly from the EVB, in the case of the former was due to strong recommendation from I^2SE. Meanwhile, in the case of the TVS diode was to ensure compatibility without the need to invest time into finding another TVS diode with the same ratings but that was in a slightly different footprint. In addition, a footprint was chosen for the TVS diode that allows many different types of diode footprints to be mounted without any compromise. Finally the LDO was chosen with a $SOT - 223 - 3$ footprint and this also gives a good compromise between space saving and power rating.

3.5.3 PCB

The Printed Circuit Board (PCB), just like with all PCBs, had to follow a certain set of parameters. All PCBs have to adhere to some sort of restrictions and these restrictions obviously shape the overall design of the PCB. Some designers might be lucky enough to not have any restrictions at all but they would still be restricted by the laws of physics. The largest contributing factor to these restrictions is usually cost of manufacturing or the cost of taking up more space. For example a lot of PCB manufacturers can drill holes as small as 0.1mm, but since any holes smaller than 0.3mm increase the cost of production sometimes by more than twice the original amount, it would mean anyone designing a board at cost would need to use holes bigger than 0.3mm. This iwas also the case for the *HomePlug GP module* developed in this thesis

The HomePlug GP module PCB had to be 22mm by 69mm.
This is was simply because that the maximum space available in the chassi of the product. The prototype EVSE as a whole package has a selling point of fitting in the same metal box that traditionally houses a two outlet block heater. Except of course the prototype EVSE can also charge electric vehicles. Since the *stamp micro* is 22mm wide and the PCB had to be as small as possible, it meant that it could neither be wider nor narrower than 22mm. The same goes for the 69mm of length, it was just large enough that the module as a whole could be plugged into the main board of the EVSE without hitting the larger components that are present there.

The PCB was not allowed to have more than four layers.
Similarly to the hole issue, increasing the number of layers to more than four has an exponential impact on the price. Additionally, since the routing would not have worked out on two layers, the number of layers was pretty much decided upon immediately.

As many components as possible are populated on the same side.
When the PCBs have finished production, they are sent to the Surface Mount Technology (SMT) assembly where the components are placed on the PCB and soldered on. If only one side of the PCB has components on it then the SMT process is much simpler and hence the cost is kept lower. If the PCB is to be assembled on both sides the process becomes more complicated after the first side has been soldered. The PCB is assembled from the top so when one side is assembled, it cannot lay flat to assemble the second side. Hence more sophisticated machines are required to assemble the second side. Additionally, when the components are being soldered on, they are put through a big oven that melts the solder paste the components are sitting on and then the PCBs are cooled down and the solder hardens and the components sit rigid. However, when soldering the second side, the manufacturer has to make sure that the bottom side of the PCB does not also heat up since then the components on the bottom side will come loose and fall off. Therefore, single sided assembly is less costly. This precaution is taken in case this PCB is to be mass produced in the future.

As little routing as possible on the outer layers.
Patenting hardware, especially circuits is very difficult but you can make it a lot harder for your competitors to copy you or to Figure what you are doing. Therefore if very little tracing is placed on the outer layers then it is more difficult to discern with the naked eye what component is connected to what. This means that competitors have to try harder to try and Figure out what your PCB does. For example to see the the traces on the inner layers of a PCB one has to invest in an X-ray machine or meticulously test all combinations of pads and pins and somehow document this.

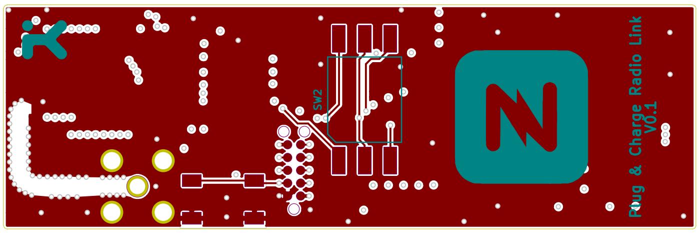


Figure 18: Top layer of the PCB

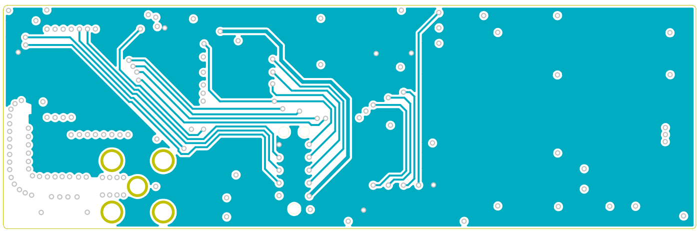


Figure 19: First inner layer of the PCB. This was also the 3.3V power plane

With these instructions and restrictions in mind, the PCB was set out to be designed. The final result can be seen in Figures 18 to 23.

The PCB was then ordered to be manufactured. After two three weeks of patient waiting it arrived in the post and the real PCB can be seen in Figure 24 and 25.

3.6 Testing of the final PCB

After the PCB arrived, it was assembled to a bare minimum, meaning the components to run the *radio IC* and the *radio IC itself* were not mounted, so that it could be tested. Only the *stamp micro* and the necessary parts needed to power the *stamp micro* were assembled on the PCB. The reason for this is because the code that runs on the *Raspberry Pi* had not been ported to run on the *Radio IC* yet so it was simplest to just connect the *HomePlug GP module* that was built with a *stamp micro* to a *Rraspberry Pi* and run it in the same environments as the *stamp micro EVB*.

Due to the timing of the PCBs arriving, there was no PEV available so the built PCB could

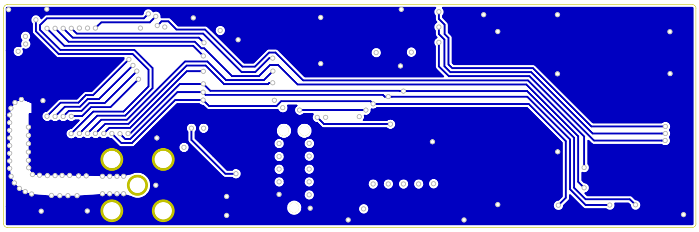


Figure 20: Second inner layer of the PCB

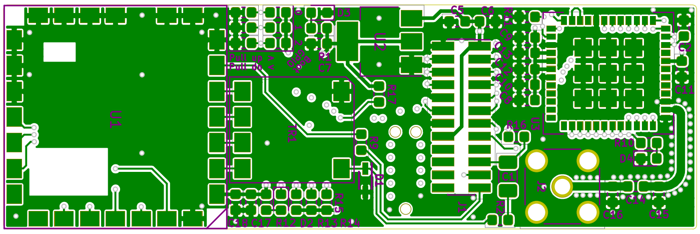


Figure 21: Bottom Layer of the PCB. (Mirrored)

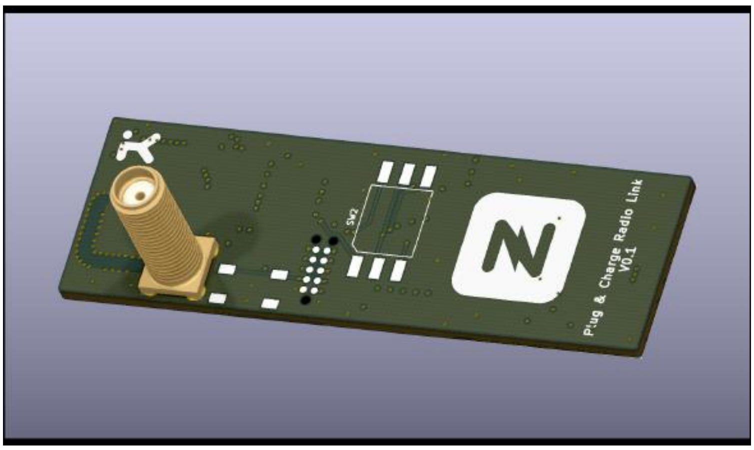


Figure 22: A 3D render of the PCB seen from the top

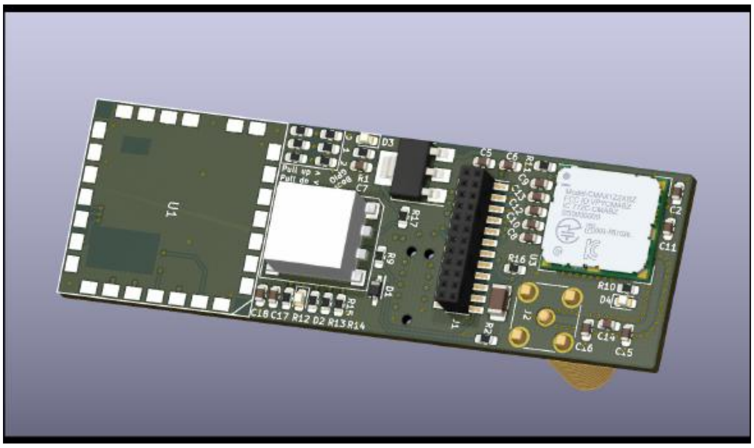


Figure 23: A 3D render of the PCB seen from the bottom. The *stamp* micro can not be seen here since there was no 3D model of it, however, the *Radio IC* can bee seen on the right side

only to be tested in a desk environment, the one seen in Figure 10. However, it is assumed that if the *HomePlug GP module* behaves the same way as the EVB did in Figure 10 then it is going to behave the same way in the other configurations as well.

After running the *HomePlug GP Module* in the same configuration as seen in Figure 10 as seen in Figure 26. When running the software the following result was obtained,

- Matching process begins
- Matching Process succeeds
- Charging starts
- Charging ends after predetermined time
- Connection is closed
- EVSE starts listening for next connected PEV.

Which means this was a success. The *HomePlug GP module* has been implemented correctly. Now all that remains to be done is to port the C code that is running on the Raspberry Pi micro controller to run on the *Radio IC* which unfortunately is outside the scope of this thesis.

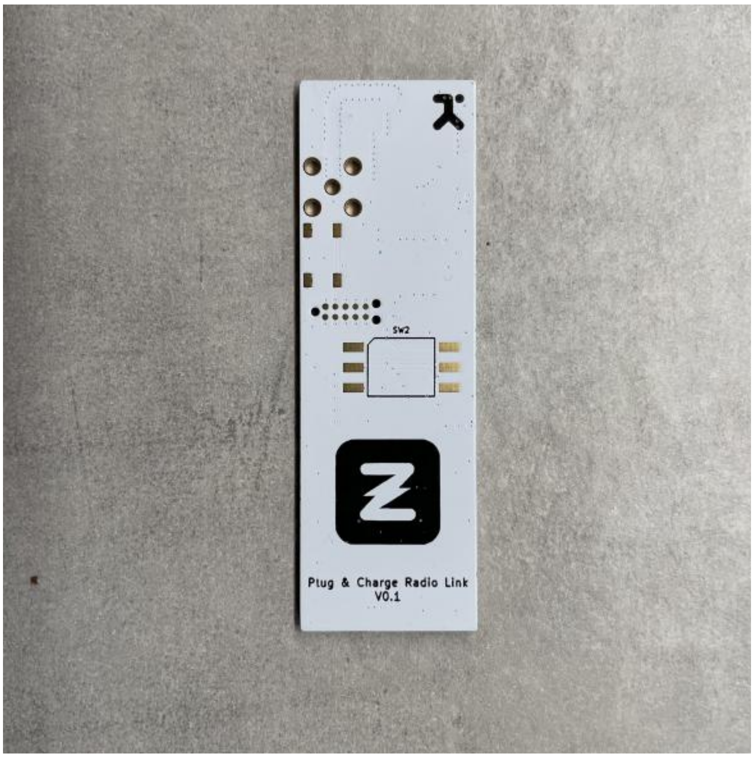


Figure 24: The real PCB seen from the top

4 Result

What was achieved in this thesis was the implementation of the *PLC Stamp Micro 2* into a module that as a contained unit can be a modular part of not only the Prototype EVSE but any J1772 compliant EVSE on the market. This means that after dropping this into the J1772 compliant EVSE it also becomes HomePlug Green PHY compliant. This is of course assuming that the *Radio IC* is correctly programmed.

Thanks to the electronics market there are some components that are interchangeable as well without modifying the PCB. As an example, the LDO can be swapped out for another that has the same foot print and pin out but can handle input voltages as high as $45V$. This is of course true for several other components on the board as well. Making the solution easily modifiable for future solutions.

Lastly, the HomePlug Green PHY module, since it complies with the standard and due to it, in fact, being a module means that it can also be compatible forthcoming EVSE generations. Even if all the other parts of the EVSE change, the future engineer who develops the forthcoming version of the EVSE can just include a connector and the appropriate necessary signals needed and automatically have HomePlug GP compliance ready.

The only drawback with the design that is clearly evident is the lack of a circuit that generates a zero cross signal. While it might seem beneficial to have $230VAC$ separated on a different PCB somewhere else, and the author does see the benefit of this, it means whatever EVSE this HomePlug GP module is being connected to has to have some means to generate a zero cross signal.

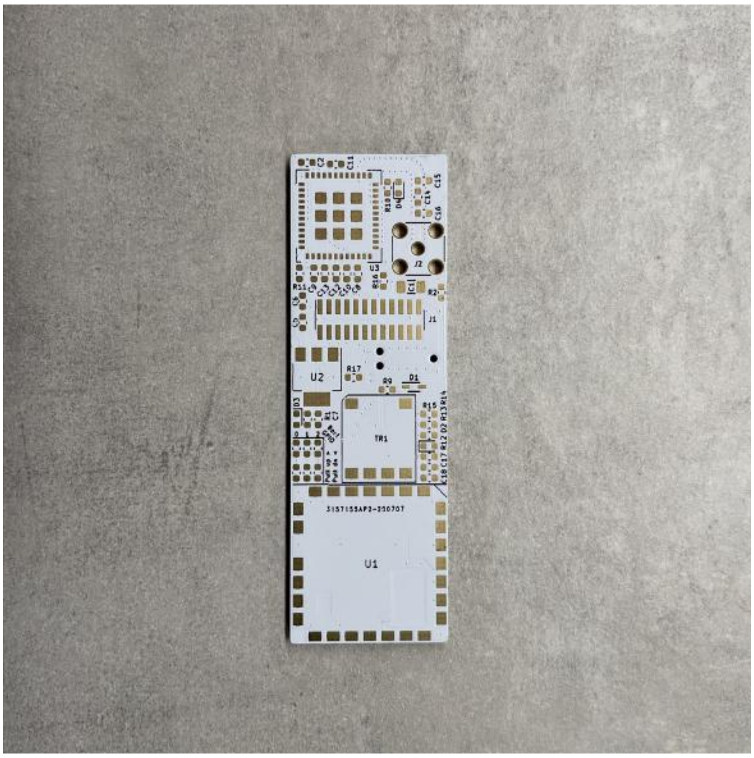


Figure 25: The real PCB seen from the bottom

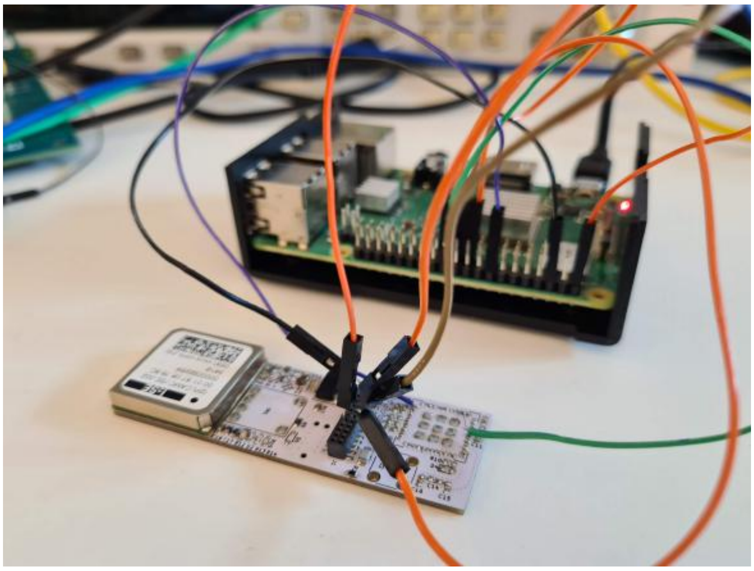


Figure 26: Assembled PCB connected to a Raspberry Pi micro controller . There is another Raspberry Pi micro controller emulating a PEV that is not seen in the picture.

5 Discussion

5.1 How well does the end product achieve the initial goal?

The end product is a PCB that not only integrates well with the prototype EVSE but with few or no modifications can integrate with most J1772 compatible EVSE out there.

5.2 Are there better solutions?

5.2.1 HomePlug AV modem

A "Down tuned" HomePlug AV modem can act as HomePlug GP modem and this opens up more hardware/IC options. This way is not limited to Qualcomm solution but a custom solution can be made "chip up" to follow the HomePlug GP standard even though the chip is HomePlug AV compliant. While this does have some benefits such as better integration with the EVSE due to it giving more freedom for the engineers, it might not be completely possible.

Even though HomePlug GP is a simplified version of HomePlug AV, it does have a few unique things about it as discussed in the theory section of this thesis. For example the SLAC process is unique to HomePlug GP and it is unclear if that is possible on a HomePlug AV modem.

5.2.2 CHAdeMO

While the CHAdeMO charging standard has its own benefits and drawbacks, it does one thing that in the opinion of the author is better. Instead of creating a whole new communication standard, it simply uses the CAN Bus. [1]

The Controller Area Network bus has been ubiquitous in the automotive industry since the 1990's and it is not only mandatory by regulation but also very flexible. Due to it being in use for so long there are several encryption layers developed on top of the CAN bus that it offer just as robust security as the HomePlug communication protocol which is ethernet package based. CAN bus solutions are cheap and easy to come by and in allot of cases would require no additional hardware from the PEV perspective. This is simply because there is a very good chance that a PEV already has a CAN bus compatible Electronics Control Unit (ECU) somewhere close to the charging port that it can just connect to. There would not be any need for identification and association processes such as SLAC due to identification being built into the CAN interface, even if dozens of PEVs connect to the same CAN Bus.

Of course there would be drawbacks to a CAN solution. It would require at least two additional wires and connectors in the charging cable. However that is not absolutely necessary. For example a system could be developed where if the EVSE and the PEV are both able to communicate with this higher standard, then there is no need for the Proximity Pilot signal nor for the Control Pilot signal so those two conductors could be used for CAN communication.

References

[1] CHAdeMO Association. *CHAdeMO Technology Overview.* Web article, 2013. https://www.chademo.com/technology/technology-overview/.

[2] Multiple Authors. *open-plc-utils*, 2014. https://github.com/qca/open-plc-utils.

[3] Circuitbasics. Basics of the spi communication protocol. Online article, May 2020.

[4] International Organization for Standardization. *Road vehicles – Vehicle to grid Communication interface. Part 3: Physical and data link layer requirements* . Standards Document. ISO 15118-3:2016, SIS - Swedish Standards Institute, April 2016.

[5] International Organization for Standardization. *Road Vehicles – Vehicle to grid communication interface. Part 1: General information and use-case definition* . Standards Document. ISO 15118-1:2019, SIS - Swedish Standards Institute, May 2019.

[6] I2SE GmbH. *PLC Stamp micro EVB Data Sheet* , February 2017. `https://in-tech-smartcharging.com/assets/Downloads/datasheet_plcstampmicro_evb_rev5-(1).pdf`.

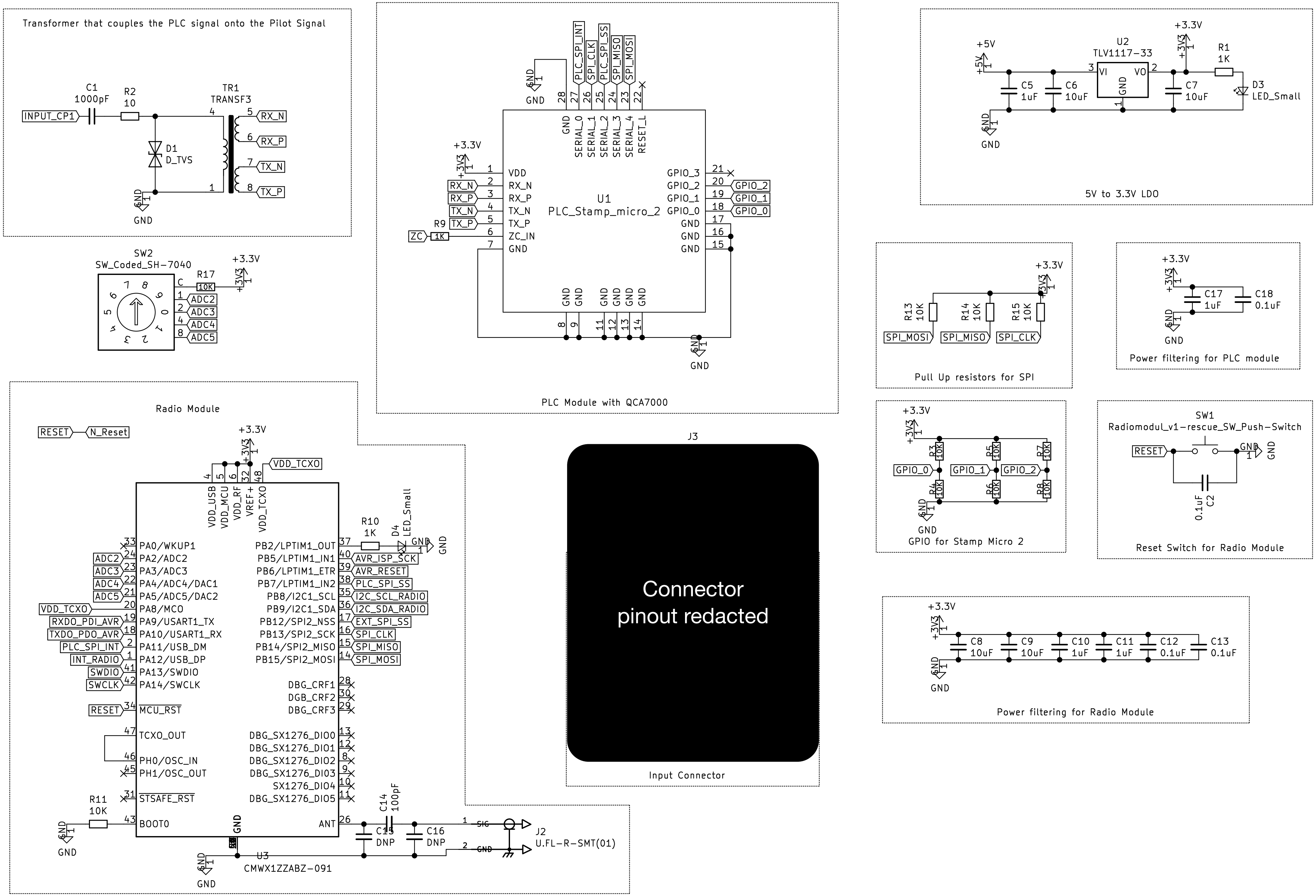
[7] I2SE GmbH. *PLC Stamp micro 2 Data Sheet* , September 2017. `https://in-tech-smartcharging.com/assets/Downloads/datasheet_plcstampmicro2_rev9.pdf`.

[8] Inc HomePlug Powerline Alliance. *Home Plug Green PHY, The Standard For In-Home Smart Grid Powerline Communications.* Whitepaper, June 2010. `https://content.codico.com/fileadmin/media/download/datasheets/plc-homeplug-green-phy/homeplug_green_phy_whitepaper.pdf`.

[9] Texas Instruments Incorporated. *SAE J1772-Compliant Electric Vehicle Service Equipment Reference Design for Level 1 and 2 EV Charger* . Design guide, October 2019. Design Guide: TIDA-010071.

[10] Ltd. Murata (China) Investment Co. *Sub-G Module Data Sheet* , October 2018. `https://wireless.murata.com/pub/RFM/data/type_abz.pdf`.

A Schematic Sheet for the module



Transformer that couples the PLC signal onto the Pilot Signal
C1
1000pF
R2
10
TR1
TRANSF3
INPUT_CP1
RX_N
RX_P
TX_N
TX_P
D1
D_TVS
GND
PLC Module with QCA7000
U1
PLC_Stamp_micro_2
PLC_SPI_INT
SPI_CLK
PLC_SPI_SS
SPI_MISO
SPI_MOSI
SERIAL_0
SERIAL_1
SERIAL_2
SERIAL_3
SERIAL_4
RESET_L
VDD
ZC_IN
GPIO_3
GPIO_2
GPIO_1
GPIO_0
R9
1K
ZC
+3.3V
5V to 3.3V LDO
U2
TLV1117-33
+5V
C5
1uF
C6
10uF
C7
10uF
R1
1K
D3
LED_Small
SW2
SW_Coded_SH-7040
R17
10K
ADC2
ADC3
ADC4
ADC5
Pull Up resistors for SPI
R13 10K
R14 10K
R15 10K
Power filtering for PLC module
C17
1uF
C18
0.1uF
GPIO for Stamp Micro 2
R3 R4 R5 R6 R7 R8 10K
Reset Switch for Radio Module
SW1
Radiomodul_v1-rescue_SW_Push-Switch
RESET
C2
0.1uF
Radio Module
N_Reset
VDD_TCXO
U3
CMWX1ZZABZ-091
VDD_USB
VDD_MCU
VDD_RF
VREF+
VDD_TCXO
PA0/WKUP1
PA2/ADC2
PA3/ADC3
PA4/ADC4/DAC1
PA5/ADC5/DAC2
PA8/MCO
PA9/USART1_TX
PA10/USART1_RX
PA11/USB_DM
PA12/USB_DP
PA13/SWDIO
PA14/SWCLK
MCU_RST
TCXO_OUT
PH0/OSC_IN
PH1/OSC_OUT
STSAFE_RST
BOOT0
PB2/LPTIM1_OUT
PB5/LPTIM1_IN1
PB6/LPTIM1_ETR
PB7/LPTIM1_IN2
PB8/I2C1_SCL
PB9/I2C1_SDA
PB12/SPI2_NSS
PB13/SPI2_SCK
PB14/SPI2_MISO
PB15/SPI2_MOSI
DBG_CRF1
DGB_CRF2
DBG_CRF3
DBG_SX1276_DIO0
DBG_SX1276_DIO1
DBG_SX1276_DIO2
DBG_SX1276_DIO3
SX1276_DIO4
DBG_SX1276_DIO5
ANT
RXDO_PDI_AVR
TXDO_PDO_AVR
INT_RADIO
SWDIO
SWCLK
AVR_ISP_SCK
AVR_RESET
I2C_SCL_RADIO
I2C_SDA_RADIO
EXT_SPI_SS
R10
1K
D4
LED_Small
R11
10K
C14
100pF
C15
DNP
C16
DNP
J2
U.FL-R-SMT(01)
SIG
J3
Connector
pinout redacted
Input Connector
Power filtering for Radio Module
C8 10uF
C9 10uF
C10 1uF
C11 1uF
C12 0.1uF
C13 0.1uF

B Extract from [4] ISO 15118-3:2016

Table A.1 *(continued)*

Parameter	Description	Min	Typical	Max	Unit
TP_match_sequence	General performance time for subsequent requests after a response to previous request has been received			100	ms
TT_amp_map_exchange	Timeout timer that runs on both EV and EVSE side after link is detected If an EV or EVSE does not want to start an AMP MAP Exchange and no request is received by the other side within the timeout value of this timer, then it is to be assumed that no AMP MAP Exchange will take place and a D-LINK_READY.indication is to be sent to the HLE.			200	ms
TT_EV_atten_results	Time the EV shall wait for CM_ATTEN_CHAR.IND messages from EVSEs. Timer starts with the sending of the first CM_START_ATTEN_CHAR.IND			1 200	ms
TT_EVSE_match_MNBC	Timeout on the EVSE side that triggers the calculation of the average attenuation profile	600	600		ms
TT_EVSE_match_session	Maximum time from the expiration of TT_EVSE_match_MNBC and the reception of either CM_VALIDATE.REQ or CM_SLAC_MATCH.REQ			10	s
TT_EVSE_SLAC_init	Time between detecting state B and receiving CM_SLAC_PARM.REQ on the EVSE side	20		50	s
TT_match_join	Maximum time between CM_SLAC_MATCH.CNF and link establishment If there is no link after this timeout expires, EV retries matching process and EVSE resets its state machine			12	s
TT_match_response	Time that the EV/EVSE shall wait for a response from the EVSE/EV			200	ms
TT_match_sequence	Time that the EVSE/EV shall wait for a request from the EV/EVSE			400	ms
TT_matching_repetition	Time duration for repetitions of the matching process when an error occurs	10			s
TT_matching_rate	Time to wait for a repetition of the whole matching process after a failed matching process	400			ms

A.9 Matching EV – EVSE process

[V2G3-A09-01] The matching process shall base on the messages defined in *[HPGP]*.

[V2G3-A09-02] The configuration of the low-layer communciation module as described in A.4.5 shall be done prior to entering the matching process.

Figures A.1and A.2 outlines the complete sequence of the matching process. It shows the sequence to follow, from the discovery of the other low-layer communciation modules to the start of the nominal communication.

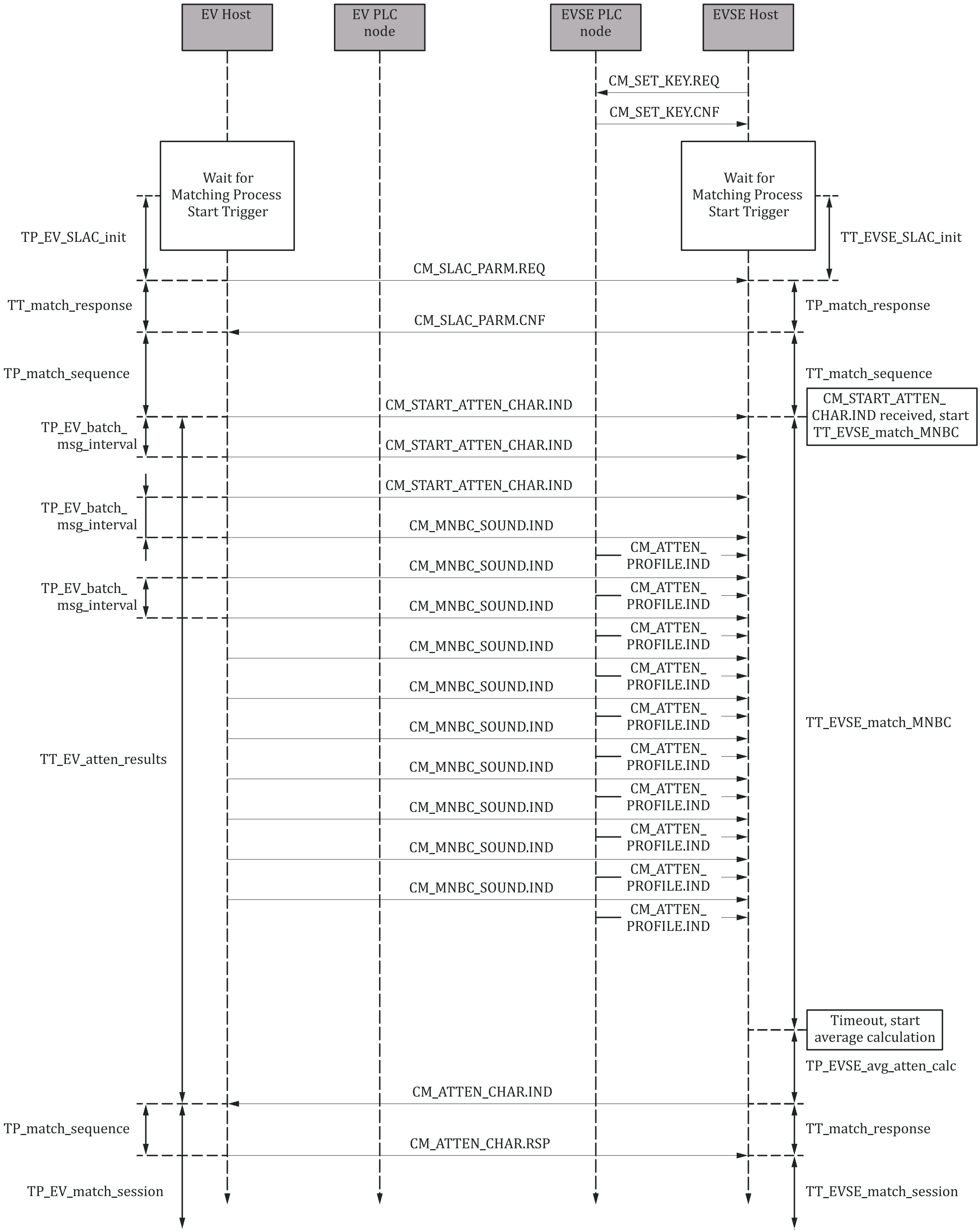


Figure A.1 — Sequence chart of HomePlug Green PHY matching process (part 1)

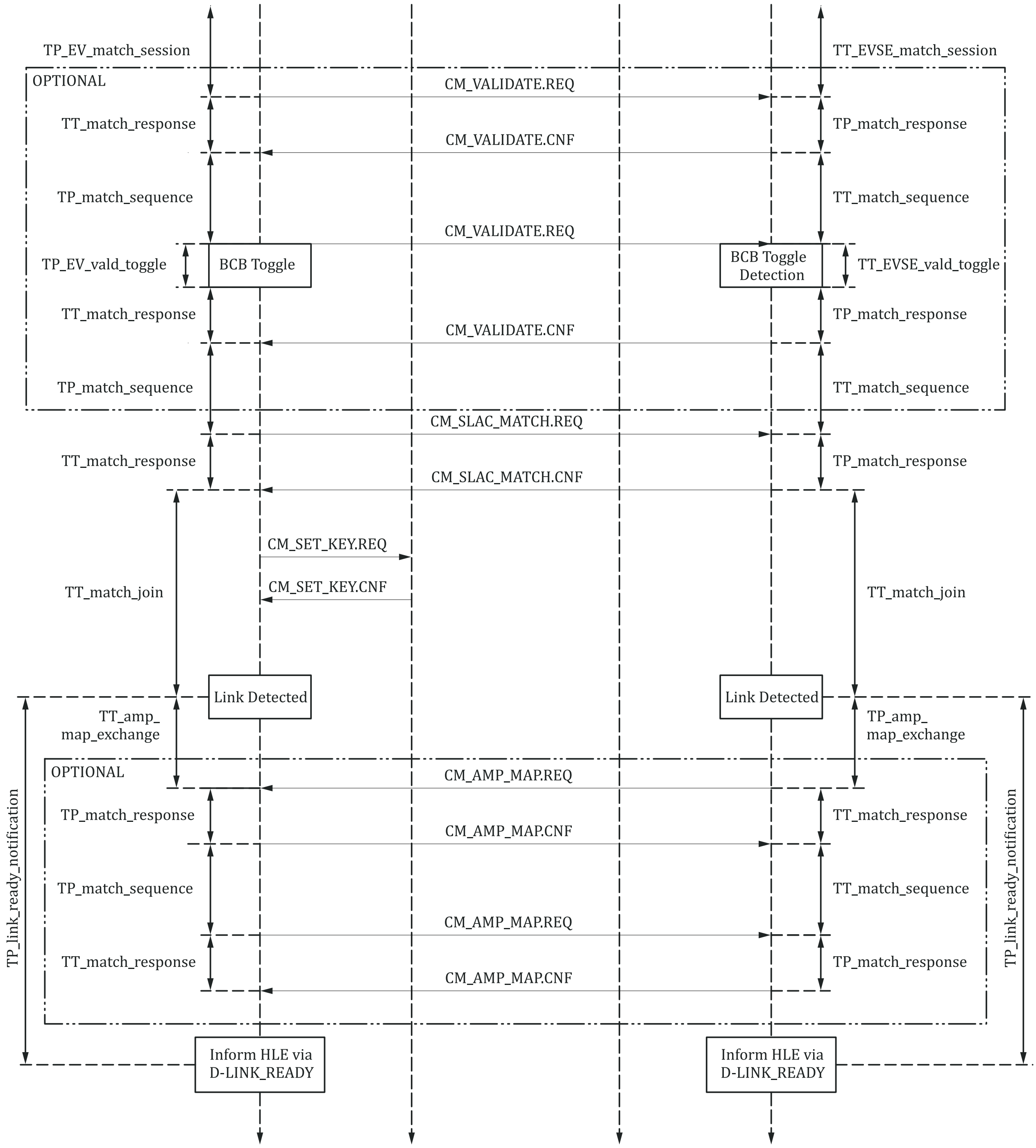


Figure A.2 — Sequence chart of HomePlug Green PHY matching process (part 2)

A.9.1 Signal strength measurement – parameter exchange

A.9.1.1 Functional description

Before the signal strength measurement starts, the EV broadcasts the parameters to be used for the following signal strength measurement sequence by means of the message CM_SLAC_PARM.REQ. Any unmatched EVSE which receives the parameter exchange broadcast sends a response to the EV by means of the message CM_SLAC_PARM.CNF.

[V2G3-A09-03] A SLAC request shall only be responded by EVSEs low-layer communciation module if all the following conditions are fulfilled:

— EVSE is connected to an EV, detected by a valid control pilot;

— EVSE is in "Unmatched" state.

A.9.1.2 Description of involved MMEs

[V2G3-A09-04] The MMEs for the parameter exchange shall be used with the following content and be sent as Ethernet unicast/broadcast messages as defined.

Table A.2 — Involved MMEs for parameters exchange

MME	Field	Octet No.	Field Size (Octets)	Value	Definition
CM_SLAC_PARM.REQ Broadcast	APPLICATION_TYPE	0	1	0x00	Fixed value indicating "PEV-EVSE matching"
	SECURITY_TYPE	1	1	0x00	Fixed value indicating "No Security"
	RunID	2 to 9	8	variable	Identifier for a matching run, randomly chosen by the EV for each CM_SLAC_PARM.REQ message and constant for all following messages of the same run
CM_SLAC_PARM.CNF Unicast	M-SOUND_TARGET	0 to 5	6	0xFFFFFFFFFFFF	Fixed value indicating that M-Sounds to be sent as Ethernet broadcast
	NUM_SOUNDS	6	1	C_EV_match_MNBC	Number of expected M-Sounds transmitted by the GP station during the SLAC process
	Time_Out	7	1	0x06	Duration TT_EVSE_match_MNBC while the EVSE receives incoming M-SOUNDS after a CM_START_ATTEN_CHAR.IND
	RESP_TYPE	8	1	0x01	Fixed value indicating "Other GP station"
	FORWARDING_STA	9 to 14	6	EV Host MAC address	The destination of SLAC results is always the EV Host
	APPLICATION_TYPE	15	1	0x00	Fixed value indicating "PEV-EVSE Matching"
	SECURITY_TYPE	16	1	0x00	Fixed value indicating "No Security"
	RunID	17 to 24	8	variable	This value shall be the same as the one sent in the CM_SLAC_PARM.REQ message by the EV