Misalignment Tolerant Inductive Power Transfer (IPT) Systems

WP-1: Sensing and wireless communication

Authors:
Keith Klein 4298292
Minh Hung Dúóng 4206584

Delft University of Technology
as part of the BSc graduation project

Jianning Dong, Supervisor
Soumya Bandyopadhyay, Supervisor

Delft, TU Delft, August 1, 2017

Abstract

This Document contains a detailed technical report on the design and test of a system that performs data acquisition and wireless communication. The system is specifically designed for an Inductive Power Transfer (IPT) charging system consisting of two control systems implemented in order to achieve maximum efficiency. One on the primary side of the charging system, and one on the secondary side. The data acquisition system is designed to measure high voltage and current for the secondary control system, and the wireless communication system is designed to calculate and transmit the set-point for the primary control system. The design includes a selection of components for each subsystem, as well as circuits and block diagrams for the implementation. A derivation of the set-point for the primary controller is also included. The system is designed for an MCU and the modules are tested using an experimental prototype.

Contents

1 Problem definition

1.1 Problem scope

The popularity of electric vehicles has significantly increased over the past years and more electric vehicles are emerging on the roads[1]. These vehicles charge their battery by connecting them to an EV charging station through the use of a cable. To make the charging process more convenient, wireless charging through Inductive Power Transfer (IPT) can be used.

An IPT wireless charging system consists of 2 coils. One coil is located on the ground while another coil is mounted on the car. The coils can be seen in figure 1.1. By applying alternating voltage on the primary coil (the coil on the ground), a current will flow through the secondary coil (the coil on the car) when the coils are close to each other. This current will charge the battery in the electric vehicle.

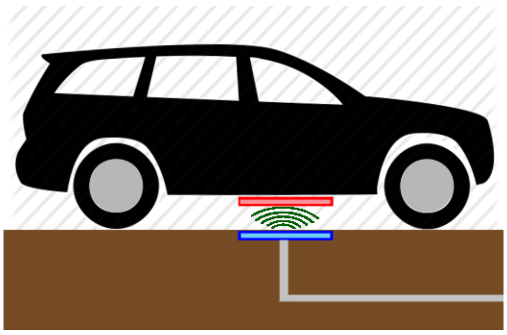


Figure 1.1: IPT wireless charging system

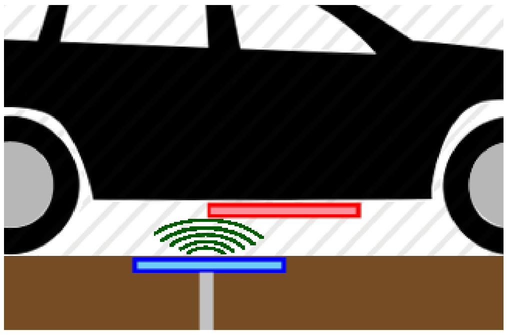


Figure 1.2: Example of misalignment

Of course, there are problems when electric vehicles are charged through the use of an IPT wireless charging system. One of these problems is misalignment, where the 2 coils are not positioned precisely above each other. This can be seen in figure 1.2. Due to the misalignment, there are more losses in the system. In order for the IPT wireless charging system to charge with maximum efficiency, a system should be designed and implemented to compensate for the misalignment effects. The design of this system will be discussed in this report.

1.2 Technical review

The total IPT charging system with the compensation system for the misalignment effects can be seen in figure 1.3. The charging system contains an inverter, a transformer, a rectifier and capacitors. The capacitors in the system are used to compensate for the inductance of the transformer.

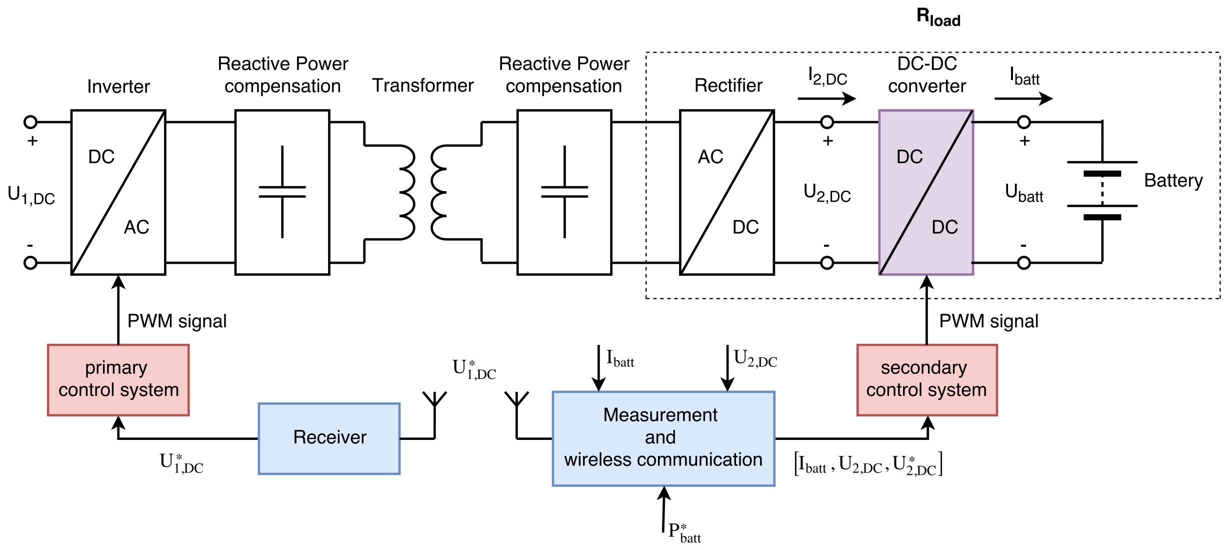


Figure 1.3: This figure shows a block diagram of the entire IPT charging system. The colored blocks represent the different parts of the misalignment compensation system that will be designed by different subgroups.

Other components are also needed for an IPT charging system that can charge with maximum efficiency. These components can be divided into 3 sections.

- The DC-DC converter
- The control system
- The measurement and wireless communication module

The rectifier, DC-DC converter and battery in figure 1.3 can be grouped together and be seen as a resistor R_{load}, which will be calculated later in the section 2.4.1. The DC-DC converter is used to change the value of R_{load}, and the control system controls the output of the DC-DC converter and the inverter.
The measurement and wireless communication module has the job of measuring the current I_{batt} (the battery current) and the voltage $U_{2,DC}$ (the voltage after the rectifier). The module

then has to pass these values to the control system on the secondary side of the charging system, the side with the battery. The measurement and wireless communication module also has the job of sending the value of the impedance matched set point $U^*_{2,DC}$ to the secondary control system. This impedance matched set point $U^*_{2,DC}$ is the value that $U_{2,DC}$ needs to have to get the target load resistor R_{load}, which will let the charging system work with maximum efficiency. The calculation of the set point will be explained in section 2.4.1.
The control system on the primary side of the charging system, the side with the inverter, also needs the value for an impedance matched set point. This control system needs the set point for $U_{1,DC}$, the input voltage of the charging system. In order for the primary control system to receive this set point value, wireless communication has to be used by the measurement and wireless communication module. The calculation of this set point will also be explained in section 2.4.1.

The DC-DC converter and the control system modules will not be discussed in this report and will be developed by other subgroups. The main focus of this report is the design and implementation of the measurement and wireless communication module.

1.3 Design requirements

The end product for the measurement and wireless communication module has to meet a few design requirements that were given by our supervisor S. Bandyopadhyay. If these requirements are met, then the end product can be used for the IPT wireless charging system. The design requirements are:

- The current sensor can measure the battery current in the range of $0 \leq I_{batt} \leq 5$ A and with an accuracy of 0.01 A.
- The voltage measurement circuit can measure the rectifier voltage in the range of $0 \leq U_{2,DC} \leq 20$ V and with an accuracy of 0.01 V.
- the formula for the impedance matched set points $U^*_{1,DC}$ and $U^*_{2,DC}$ must be implemented on the control board
- The primary control system must receive the set point $U^*_{1,DC}$ with an error rate of at most 1% through wireless communication over a distance of 1 m.

2 Design description

2.1 Overview

The overview of the measurement and wireless communication system can be seen in figure 2.1.

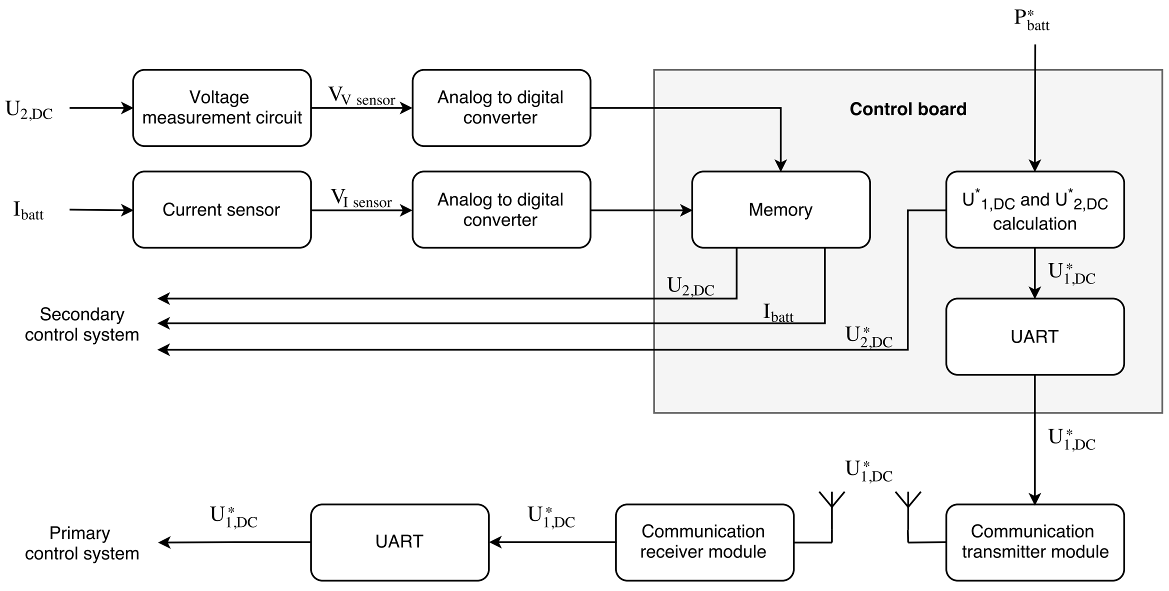


Figure 2.1: This figure shows the block diagram of the measurement and wireless communication system.

This system can be divided into 2 parts. The first part is the measurement and data handling part that will be discussed in section 2.3. It contains a voltage measurement circuit (section 2.3.2) and a current sensor (section 2.3.3) to measure the rectifier voltage $U_{2,DC}$ and the battery current I_{batt}, respectively. These values will go through the analog-to-digital converters (section 2.3.1) and be stored on the memory of the control board. The choice for the type of control board will discussed in section 2.2.

The second part of the measurement and wireless communication system (section 2.4) is the part that handles the calculation of the impedance matched set points $U^*_{1,DC}$ and $U^*_{2,DC}$. It also has the job of sending the set points to the control systems. In order for the primary control system to receive $U^*_{1,DC}$, wireless communication is needed. The calculation of the impedance matched set points will be discussed in section 2.4.1, and the transmission of $U^*_{1,DC}$ through wireless communication in section 2.4.3. The UART (universal asynchronous receiver/transmitter) in figure 2.1 is the component that handles the wireless transmission of the set point. Before this value can be transmitted, the wireless communication protocol has to be chosen first. The choice for the wireless communication protocol will be discussed in section 2.4.2.

So the inputs of the measurement and wireless communication system are the voltage $U_{2,DC}$ and the current I_{batt}. However, figure 2.1 shows that there is another input signal. This signal is the target battery power P^*_{batt}. The input P^*_{batt} is needed for calculation of the impedance matched set points and will be given be given by our supervisor.
The outputs of the measurement and wireless communication system are also shown in figure 2.1. As it can be seen, there are 4 output signals. 1 output signal, $U^*_{1,DC}$ will go to the primary control system that was explained earlier. The other 3 output signals will go to the secondary control system. The output signals are the voltage $U_{2,DC}$, the current I_{batt} and the set point $U^*_{2,DC}$.

2.2 Control board

There are two types of integrated circuit(IC) types that can be used as a control board for this application. Namely, a Field Programmable Gate Array (FPGA) or a Micro Controller Unit (MCU). This section will discuss the advantages and disadvantages of both IC types for this application, and will explain the choice made for the design and implementation.

2.2.1 FPGA

FPGA boards consist of an array of programmable switches that allow specific logic circuits to be connected using a Hardware Description Language (HDL) in order to implement the desired circuit [2] [3].

Advantages
Unlike Micro Controllers, FPGAs allow all inputs to be processed in parallel. For this application, all subsystems would operate simultaneously (e.g. sensing and wireless communication), which allows for a fast system. Since FPGAs do not have a fixed hardware structure, an implementation on FPGA allows the design to be optimized for any specific application. Also, because the control system will be implemented on an FPGA, it would be advantageous to

implement sensing, data handling and wireless communication on the same FPGA. This would result in better overall system integration and would allow for easier implementation on an Application Specific Integrated Circuit(ASIC), if required.

Disadvantages
Due to their parallel nature, designs on FPGAs are often more complex to implement. It is also harder to test and debug designs since intermediate signals cannot be easily printed to a computer via USB. Most FPGA development boards have no Analog inputs which means that an external ADC will need to be used if an FPGA design is chosen.

2.2.2 MCU

MCUs are small programmable computers built on a PCB with built in functions like arithmetic operations, programmable memory and -I/O pins. MCUs run programs which provide sequential instructions to fixed hardware structures in order to process data from their inputs.

Advantages
Generally, MCUs are easier to program than FPGAs because their architecture is pre-designed and features like arithmetic operations are already implemented.
Since most MCUs have a print function implemented to print variables to a serial port, it is easy to test and debug designs on MCUs.
The ATmega328P MCU has a built in 10-bit ADC, which is useful in designs where analog sensors are used. It also has a built in UART and SPI/I^2C Communication modules to interface with other devices or sensors [4]. This is useful since the wireless communication system will require a UART in order to send data sequentially. The functions of a MCU could be created inside an FPGA. However, it would be fairly complex to implement.

Disadvantages
Programs are run sequentially which means that latency increases with program size. This is not the case on FPGAs. MCUs are also slower than FPGAs for applications where a large amount of inputs need to be processed in parallel.

Based on the points mentioned above, it was chosen to design a system based on an MCU control board. This makes the design significantly easier to implement, test and debug. An MCU can also easily perform arithmetic operations. This will simplify the process of calculating the impedance matched set points. However, since the control system will be implemented on FPGA, this means that the data acquisition system will be independent from the control system. However, the MCU can easily send information to the FPGA using any of the above mentioned serial interfaces. The following sections will focus mostly on an MCU based design and implementation. The MCU that will be used is a development board based on the ATmega328P. Its specifications can be seen in table 2.1.

	ATmega328P
ADC	10-bit 15Ksps
Clock frequency (MHz)	16
# of general I/O	13 digital I/O , 8 analog inputs
Other features	Built in UART, SPI, I2C and reset switch

Table 2.1: Specifications for the ATmega328P [4].

2.3 Measurement and data handling

2.3.1 Analog to digital converter

The MCU that will be used to read data from the sensors, has a built in Analog to Digital converter(ADC). However, this ADC has a resolution of only 10 bits. The measuring resolution expressed in the unit of the parameter to be measured, is given by:

$$Res = \frac{R_{meas}}{2^{Res_{bit}}} \tag{2.1}$$

Where R_{Meas} is the range of the measurement and Res_{bit} is the bit resolution of the ADC.
For a voltage measuring range of 0 - 20V, filling in equation 2.1 for a 10-bit ADC gives a resolution of approximately 0.02V. For a current measuring range of 0 - 5A, the resolution would be approximately 0.005A. These values are not sufficient to meet the design requirements set in chapter 1. This means that an external, higher resolution ADC will be required.
When choosing the ADC, the following parameters are taken into account:

- Conversion rate (Hz)
- Power use (W)
- Bit resolution (# of bits)
- Communication interface (Serial / Parallel)

Low power use is ideal, but is not a limitation since the power source (car battery) has a very large capacity. A bit resolution of 12 bits is selected for a good balance between accuracy and conversion speed.
There are different types of ADCs each with their own advantages and disadvantages:

- Flash ADC:
  Flash ADCs use comparators to perform parallel conversions. These are the fastest but also have the highest power usage due to the large amount of comparators needed.

- Dual slope ADC:
  A dual slope ADC measures the time it takes to charge and discharge a capacitor in order to digitalize the input signal. These converters are very accurate but much slower than flash ADCs and have a variable conversion time.

- Successive approximation ADC:
  Successive approximation ADCs compare the input voltage with all possible digital voltages. They are much faster than dual slope ADCs [5]. They are slower compared to flash ADCs but require less hardware and power. This type of ADC is the most suitable for this application since it has fairly low power consumption and uses little components while still remaining adequately fast ($\approx 2 * 10^6$ samples/s).

The component chosen for the ADC is the ADS7816. Its specifications can be seen in table 2.2.

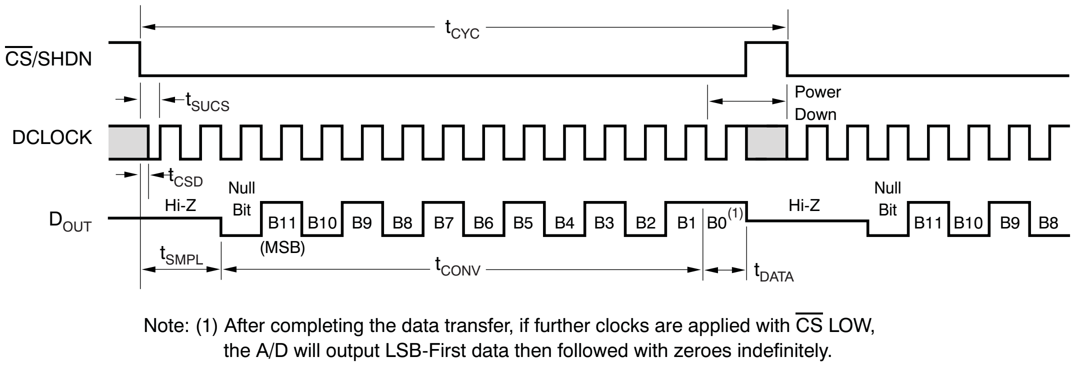
ADS7816	
Bit resolution	12
Supply voltage	5 V
Power use	1.9 mW @ 200 kHz
Input capacitance	25 pf
Communication interface	3-Wire Serial

Table 2.2: The specifications for the ADS7816 [6]

.

ADC interface

The ADCs will be directly connected to the MCU. It is possible for them to share a clock as well as a data line. However, a shared data line would halve the effective sampling frequency and would result in the requirements, set by the control system, not being met. For this reason, the ADCs will have separate data lines. Sampling of the analog input and data transfer is initiated by pulling the *CS'*(chip-select) Pin low. The data output is then initiated after the second falling edge of *DCLOCK*. The timing diagram can be seen in figure 2.2.



$\overline{CS}$/SHDN
DCLOCK
D_OUT
t_CYC
t_SUCS
t_CSD
Power Down
Hi-Z
Null Bit
B11 B10 B9 B8 B7 B6 B5 B4 B3 B2 B1 B0(1)
(MSB)
t_SMPL
t_CONV
t_DATA
Note: (1) After completing the data transfer, if further clocks are applied with CS LOW, the A/D will output LSB-First data then followed with zeroes indefinitely.

Figure 2.2: This Figure shows the timing diagram for the ADS7816[6].

Since the data output of each ADC is always initiated on the second falling edge of *DCLOCK*, it is possible to make them run synchronously by connecting the *CS'* pins of each ADC together. This regains the pin lost to the second data line and allows the ADCs to be sampled in parallel at their maximum sampling frequency. Thus, the ADCs will share *DCLOCK* and *CS'* pin, and have two separate data lines.

The amount of levels that can be output by the ADC is given by,

$$LSB = 2^{N_{BITS}}. \tag{2.2}$$

Where N_{BITS} is the bit resolution of the ADC. For the ADS7816, equation 2.2 is equal to 4096. Where 0 corresponds to 0V and 4095 corresponds to the reference voltage of the ADC.

2.3.2 Voltage measurement

Due to the fact that $U_{2,DC}$ is fairly large and the maximum input of the ADC will be limited to 5V [6], the voltage needs to be stepped down before it can be sampled by the ADC. The voltage could be stepped down using a simple voltage divider as seen in figure 2.3. However, this introduces two problems:

1. The ADS716 has a sample and hold circuit with an input capacitance of 25pF. This capacitor array must be charged to the required voltage within 1.5 clock cycles to ensure the analog value is correctly sampled [6]. If the resistors are too large, not enough current will be supplied to the ADC resulting in incorrect

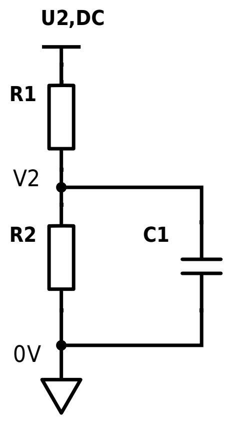


Figure 2.3: This figure shows a voltage divider. The ADC is modeled as a capacitor.

samples.

2. If the resistors are too small, a large fraction of the power will be dissipated by the resistors as heat. This not only requires physically larger resistors, but also decreases the performance of the entire system due to the temperature dependency of most resistors.

To evaluate the circuit, the following assumptions are made:

1. The ADC runs at its maximum clock rate (f_{Clk} = 3.2 Mhz). This gives a maximum allowable charge time of,

   $$T_{charge} = 1.5T_{Clk} = \frac{1.5}{f_{Clk}} = 4.69\text{e-7 s}$$

2. The ADC has no input resistance and is assumed to behave as a capacitor.
3. The sample and hold capacitor is considered full when $V2$ is 99% of its maximum voltage. Thus for a maximum voltage $V_{max} = 5V$, the capacitor is fully charged when,

   $$V2 \geq 0.99V_{max} = 4.95\text{V}.$$

4. If the absolute maximum value of $U_{2,DC}$ is assumed to be 50 V, it means that the voltage has to be stepped down by a factor of ≈10. The relationship between R_1 and R_2 becomes,

   $$R_1 = 10R_2 - R_2$$

   When testing the circuit, the voltage will be supplied by a lab bench power supply.

MATLAB was used to evaluate the circuit and plot the capacitor voltage $V2$ as a function of time for different values of R_2 with the initial condition $V2(0) = 0$. The result can be seen in figure 2.4.

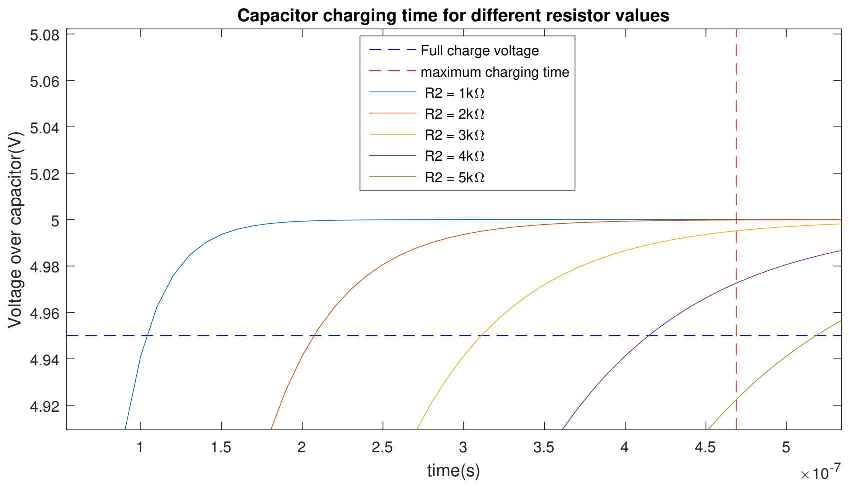


Figure 2.4: This figure shows a simulation of the voltage over the sample and hold capacitor of the ADC as a function of time.

It can be seen that the maximum usable resistance for R_2 is around $4k\Omega$. The passive power dissipation is then,

$$\frac{U_{2,DC}^2}{R_1+R_2} = \frac{U_{2,DC}^2}{R_2 * 10} = \frac{50^2}{40k\Omega} \approx 62mW.$$

While this value is not extremely large, a full scale model with a higher voltage would increase the power loss even further. The circuit can be improved by connecting an Operational Amplifier(OpAmp) as a buffer amplifier to the resistor R_2 as seen in figure 2.5. This ensures that large value resistors can be used for the voltage divider, while still providing enough output current to charge the sample and hold capacitor of the ADC quickly.
The OpAmp that was chosen for this application is the MCP6041. Its specifications can be seen in Table 2.3.

Table 2.3: The specifications for the MCP6041 [7].

Component name	**MCP6041**
Bias current	1-20 pA
Quiescent current	600 nA
Supply range	1.4-6V
Output voltage	rail to rail
Maximum output current	30 mA
Maximum capacitive load	60 pf

This OpAmp was chosen for its low noise, low quiescent current and its ability to drive high capacitive loads as a buffer amplifier.

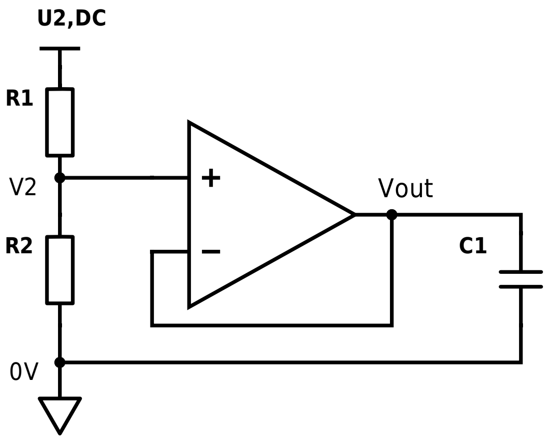


Figure 2.5: This figure shows a voltage divider connected to a buffer amplifier to drive the ADC(Represented by C1). A full schematic of the voltage measurement circuit can be seen in figure 2.6.

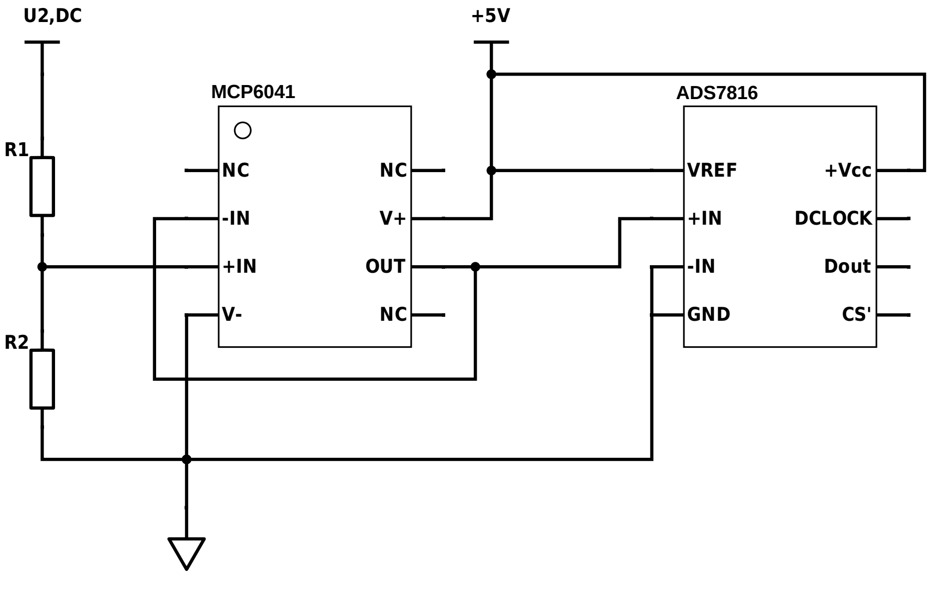


Figure 2.6: This Figure shows the schematic of the circuit that will be used to measure $U_{2,DC}$.

The ADC will be controlled using a 3 wire serial interface. *DCLOCK*, *Dout* and *CS'* will be connected to the MCU.

The values of the resistors R_1 and R_2 depend on the input resistance of the OpAmp which is determined by the bias current. As can be seen in Table 2.3, the bias current for the MCP6041 is between 1 and 20 pA. Thus, for maximum input voltage and worst case bias current, the input resistance of the OpAmp is given by:

$$R_{input} = \frac{V_{in}}{I_{bias}} = \frac{5V}{20pA} = 2.5\text{e}11\Omega = 0.25T\Omega$$

Where I_{bias} is the bias current and V_{in} is the voltage between the OpAmp terminals. The high input resistance of the selected OpAmp allows for very high resistor values for the voltage divider. The resistors that will be used will be in the order of $M\Omega$'s but the exact value will depend on the availability and the voltage range that needs to be measured.

2.3.3 Current measurement

Current sensor

There are various techniques which can be used to measure current. Some being more suitable than others for this application. The following types of sensors, which may be suitable can be identified:

1. Shunt Resistor current sensor:

As can be seen in figure 2.7, this type of sensor places a resistor along the wire of which the current needs to be measured. The voltage drop across the resistor is then measured and used to calculate the current using Ohm's law of resistance.

The advantage of this technique is that it is the simplest and cheapest way to implement current sensing, while being fairly accurate. However, this method has some problems. Since the power dissipated by a resistor increases with the square of the current,

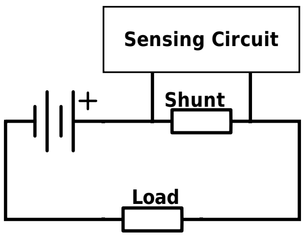


Figure 2.7

$$P = I^2R$$

It is less suitable for applications where the current that needs to be measured is large due to the heat generated by ohmic loss. Not only does this increase the power usage of the system, but since the resistance is temperature dependant, the voltage over the resistor will not be linearly proportional over the entire measuring range. This effect is reduced when using larger resistance values, but this has the effect of loading the circuit. On the other hand, small resistance values require high amplification of the output signal which may increase the complexity and cost. This type of current sensing method also provides no inherent electrical isolation between the sensing circuit and the current to be measured. The accuracy of a shunt resistor is typically in the range of 0.1-2%. Depending on the type of resistor used, its measuring range can be anywhere from mA to kA and the power loss can range from mW to kW [8].

2. Magnetic field current sensor:

This type of current sensor measures the magnetic field created by the current flowing through a conductor. Figure 2.8 shows an illustration of this principle. Since the value of the magnetic field is directly proportional to the current flowing through it, the sensor's output voltage is also proportional to the current. This principle can be used to measure the current.

The advantage of this technique is that there is inherent electrical isolation between the current measured and the sensing circuit. There are also less power losses due to the absence of a shunt resistor. The accuracy of this method ranges from 0.001% to 5% depending on the type of magnetic field sensor used. The range is in the order of mA to kA. The power loss can range from mW to W[8].

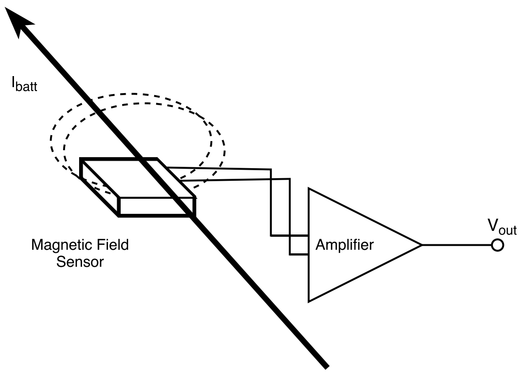


Figure 2.8: Working principle of a magnetic field current sensor. The dotted lines represent the magnetic field lines.

The sensor chosen for this application is of the second type, namely a hall-effect current sensor. This type of sensor was chosen for its low power usage, good measuring range and accuracy. They are also fairly cheap and are readily available. The sensor to be used for the prototype is the LEM LTS6-NP. Its specifications can be seen in table 2.4.

Table 2.4: This table gives an overview of the specifications for the current sensor chosen for the prototype[9].

Sensor name	**LEM LTS 6-NP**
Measuring range (A-A)	-19 - +19
Accuracy (%)	0.7%
Power consumption(mW)	150
Supply voltage (V)	5
Output Voltage (V)	0 - 5
Sensitivity (mV/A)	104.16
Interface	Analog
Type	open loop
Temperature range (Celsius)	-40 - 85
Cost (€)	13.33

The LTS6-NP will be connected to the ADC using a buffer amplifier as used in the voltage measurement circuit. This is done to ensure that enough current is supplied to the ADC, since the data-sheet does not state the output current. It is also unknown how the sensor behaves when connected to a capacitive load. The buffer amplifier will ensure that there is no load on the output of the sensor, increasing its accuracy. A schematic of the measuring circuit can be seen in figure 2.9.

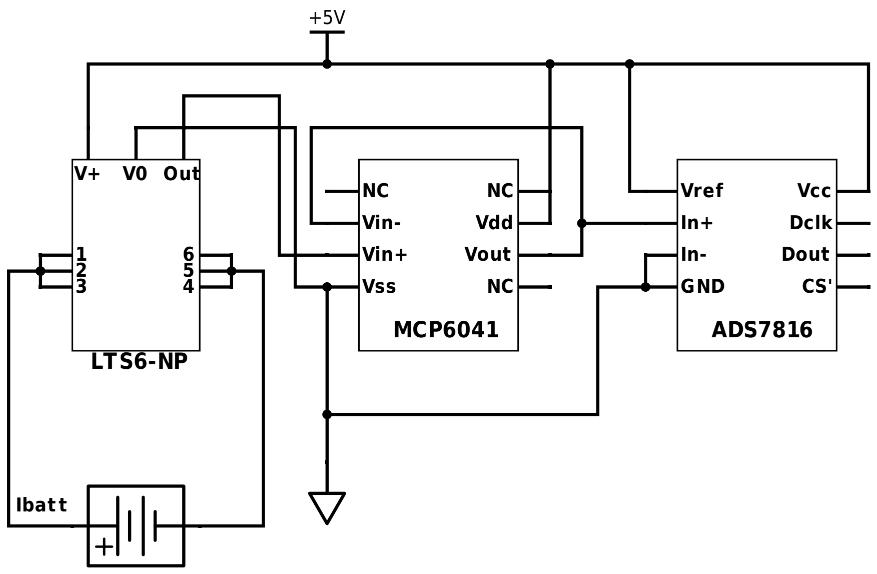


Figure 2.9: This figure shows the schematic for the current sensing measuring circuit. DCLOCK, Dout and CS' will be connected to the MCU for communication.

2.4 Wireless communication

2.4.1 Calculation of the impedance matched set points

Before calculating the impedance matched set points $U^*_{1,DC}$ and $U^*_{2,DC}$, the load resistor R_{load} (rectifier, DC-DC converter and battery) needs to be calculated first. In figure 2.10, the model for the IPT system can be seen. The voltage source $U_{1,AC}$ in the circuit is the voltage output of the inverter on the primary side of the charging system. The resistances R_p and R_s are the summed up parasitic resistances from the capacitors and coils. Resistor R_p is the parasitic resistance on the primary side of the charging system, and R_s is the parasitic resistance on the secondary side. The capacitors C_p and C_s in the figure are the used capacitors for the reactive power compensation. As it can be seen from figure 2.10, the used topology for the compensation is the serie-serie (SS) topology. The inductors in the circuit represent the coils that act as a transformer. The inductor on the primary side has the inductance L_1, and the inductor on the secondary side has the inductance L_2. The symbol M in figure 2.10 is the mutual inductance of the transformer, and this inductance can be written as $M = k\sqrt{L_1 L_2}$ where k is the coupling coefficient [10].

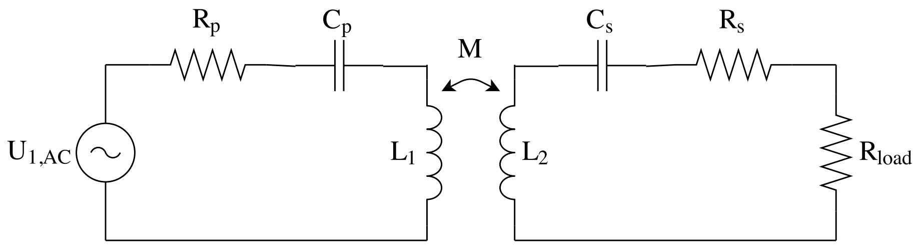


Figure 2.10: The model for the IPT system.

The IPT charging system has already been built by our supervisor, and table 2.5 shows the parameters of this IPT system.

Variable	**Value**	**Description**
ω	$2\pi 85$k rad/s	The angular frequency of the inverter
L_1	200 μH	The inductance of the primary coil
L_2	200 μH	The inductance of the secondary coil
C_p	$\frac{1}{\omega^2 L_1}$ F	The capacitance on the primary side
C_s	$\frac{1}{\omega^2 L_2}$ F	The capacitance on the secondary side
$R_p = R_s$	-	The parasitic resistances of the capacitors and coils
k	$0.08 \leq k \leq 0.20$	The coupling coefficient

Table 2.5: The IPT charging system parameters.

The 2 inductors that act as a transformer can be replaced by the equivalent transformer model, which can be seen in figure 2.11. The inductances L_p and L_s in the equivalent transformer model are the leakage inductances, and they can written as $L_p = L_1 - M$ and $L_s = L_2 - M$. With the circuit in figure 2.11, the resistance R_{load} can be derived.
The control system has the job of setting the value of R_{load} at a value that enables power transfer at maximum efficiency. The calculation of R_{load} can be seen in appendix A.1, and the result is given by equation 2.3.

$$R_{load} = M\omega\sqrt{\frac{R_s}{R_p}} = M\omega \tag{2.3}$$

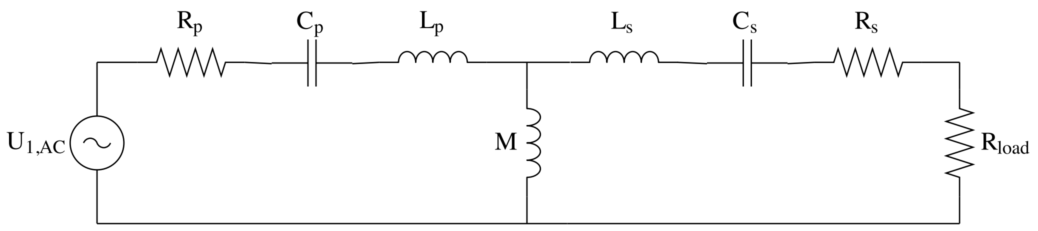


Figure 2.11: The model for the IPT system with an equivalent transformer model.

Impedance matched set point $U^*_{2,DC}$

Now that one form of R_{load} is known, another equation for R_{load} is needed first before the impedance matched set point $U^*_{2,DC}$ can be calculated. Figure 2.12 shows a circuit with an alternating current source I_{AC}, a rectifier and a load R_L. This is a model for the secondary side of an IPT charging system. The alternating current source I_{AC} represents the induced current by the primary side of the IPT system. This current is a sine wave and has an amplitude I_p.
To calculate the equivalent load resistor R_{AC} (the combined resistance of the rectifier and R_L), the equation for the load current I_0 needs to be found. I_0 can be calculated by taking the average of I_{abs}, the current after the H-bridge. The form of I_{abs} can be seen at the top of figure 2.12, and the calculation for I_0 can be seen in equation 2.4.

$$I_0 = \frac{1}{\pi}\int_0^{\pi} I_p \sin(\mathrm{t})\,\mathrm{d}t = \frac{I_p}{\pi}\left[-\cos(t)\right]_0^{\pi} = \frac{2}{\pi} I_p \tag{2.4}$$

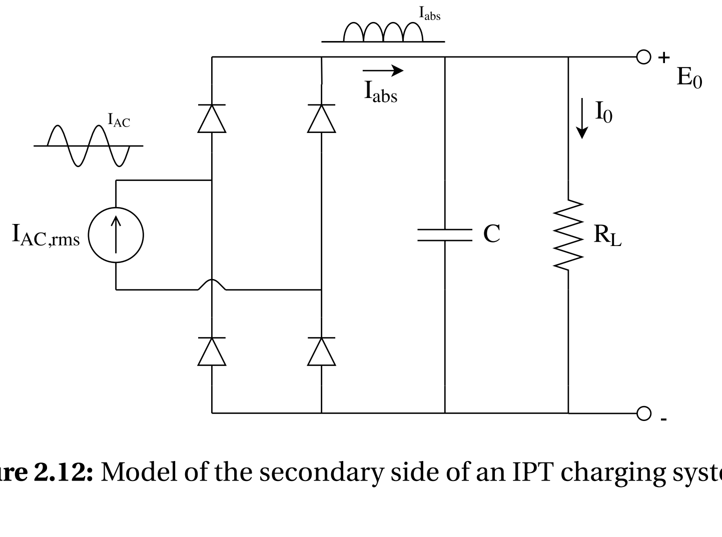


Figure 2.12: Model of the secondary side of an IPT charging system.

The amplitude I_p can be derived from the result of equation 2.4, which is:

$$I_0 = \frac{2}{\pi} I_p \Rightarrow I_p = \frac{\pi}{2} I_0 \tag{2.5}$$

The first value that is needed for the calculation of R_{AC}, is the RMS value of I_{AC} which can be seen in equation 2.6. The second needed value is $V_{AC,rms}$, the RMS value of the voltage over R_L. The equation for this voltage is given by equation 2.7 with E_0 being the load voltage.

$$I_{AC,rms} = \frac{I_p}{\sqrt{2}} = \frac{\pi}{2\sqrt{2}} I_0 \tag{2.6}$$

$$V_{AC,rms} = \frac{I_0 E_0}{I_{AC,rms}} = \frac{2\sqrt{2}}{\pi} E_0 \tag{2.7}$$

Now that $V_{AC,rms}$ and $I_{AC,rms}$ are known, R_{AC} can be found:

$$R_{AC} = \frac{V_{AC,rms}}{I_{AC,rms}} = \frac{8}{\pi^2} \frac{E_0}{I_0} = \frac{8}{\pi^2} R_L \tag{2.8}$$

The load resistor R_{load} and equivalent load resistor R_{AC} are the same. So by equating the formulas for R_{load} and R_{AC} with each other and using the equation $R_L = U_{2,DC}^2 / P_{batt}$, the inverter output voltage $U_{2,DC}$ can be calculated. The result can be seen in equation 2.9

$$\begin{aligned} R_{load} = R_{AC} \Rightarrow \omega M &= \frac{8}{\pi^2} R_L = \frac{8}{\pi^2} \frac{U_{2,DC}^2}{P_{batt}} \\ \Rightarrow U_{2,DC}^2 &= \frac{\pi^2}{8} \omega M P_{batt} \\ \Rightarrow U_{2,DC} &= \sqrt{\frac{\pi^2}{8} \omega M P_{batt}} \end{aligned} \tag{2.9}$$

So the impedance matched set point $U_{2,DC}^*$ is:

$$U_{2,DC}^* = \sqrt{\frac{\pi^2}{8} \omega M P_{batt}^*} \tag{2.10}$$

Impedance matched set point $U_{1,DC}^*$

For the calculation of the impedance matched set point $U^*_{1,DC}$, the waveform of $U_{1,AC}$ (the output voltage of the inverter) will be observed first, which can be seen in figure 2.13. The T in the figure is the period of the periodic signal $U_{1,AC}$, and the factor α indicates how wide each pulse will be. This factor can have a value between $0 \le \alpha \le \pi$.

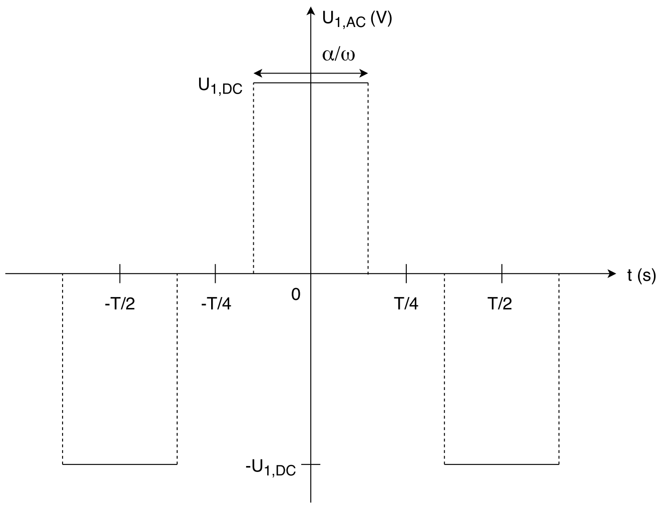


Figure 2.13: The square wave of the voltage $U_{1,AC}$.

The inverter voltage output $U_{1,AC}$ can be written as a Fourier series and the calculation of this Fourier series is explained in appendix A.2. The result can be seen in equation 2.11.

$$U_{1,AC} = \frac{4}{\pi} U_{1,DC} \sum_{k=1,3,5,...}^{\infty} \frac{\sin(k\frac{\alpha}{2})}{k} \cos(k\omega t) \tag{2.11}$$

When voltage gets transfered to the secondary side of the transformer, the side with the battery, it can be assumed that all harmonic signals and DC signals are filtered out. This is because the transformer acts as a bandpass filter when the value of the coupling coefficient is low [11]. Only the voltage with the fundamental frequency is left for $U_{2,AC}$ (The input voltage of the rectifier). So $U_{2,AC}$ is $U_{1,AC}$ with $k = 1$ divided by the turns ratio $a = \sqrt{L_1/L_2}$ of the transformer [10]. But because $L_1 = L_2$, the turns ratio in the equation disappears. The final result for the equation of $U_{2,AC}$ can be seen in equation 2.12.

$$U_{2,AC} = \frac{4}{\pi} U_{1,DC} \sin\left(\frac{\alpha}{2}\right) \cos(\omega t) \sqrt{\frac{L_2}{L_1}} = \frac{4}{\pi} U_{1,DC} \sin\left(\frac{\alpha}{2}\right) \cos(\omega t) \tag{2.12}$$

The RMS value of $U_{2,AC}$ can be seen in equation 2.13. By rewriting this equation, an expression for $U_{1,DC}$ can be obtained. This is done by moving $U_{2,rms}$ to the right of the equation and $U_{1,DC}$ to the left. By also using the equation $U^2_{2,rms} = R_{load} P_{batt}$, the final equation for $U_{1,DC}$

can be seen in equation 2.14.

$$U_{2,rms} = \frac{4}{\pi}\frac{U_{1,DC}}{\sqrt{2}}\sin\left(\frac{\alpha}{2}\right) = \frac{2\sqrt{2}}{\pi}U_{1,DC}\sin\left(\frac{\alpha}{2}\right) \tag{2.13}$$

$$U_{1,DC} = \frac{\pi}{2\sqrt{2}}\frac{U_{2,rms}}{\sin\left(\frac{\alpha}{2}\right)} = \sqrt{\frac{\pi^2}{8}\frac{U_{2,rms}^2}{\sin^2\left(\frac{\alpha}{2}\right)}} = \sqrt{\frac{\pi^2}{8}\frac{R_{load}P_{batt}}{\sin^2\left(\frac{\alpha}{2}\right)}} = \sqrt{\frac{\pi^2}{8}\frac{\omega M P_{batt}}{\sin^2\left(\frac{\alpha}{2}\right)}} \tag{2.14}$$

The used equation for the calculation of $U_{1,DC}^*$ is the one in equation 2.15. To get equation 2.15, the factor $P_{batt}/\sin^2\left(\frac{\alpha}{2}\right)$ in equation 2.14 is replaced by P_{batt}^*, the target battery power.

$$U_{1,DC}^* = \sqrt{\frac{\pi^2}{8}\omega M P_{batt}^*} \tag{2.15}$$

Both the set points $U_{1,DC}^*$ and $U_{2,DC}^*$ will be stored on the control board as single precision floating points, which will be accurate enough for the control system.

2.4.2 Wireless communication protocol

For the transmission of the impedance matched set-point $\mathrm{U}_{1,DC}^*$ to the control system on the primary side, different wireless communication protocols can be used. The available protocols are [12] [13]:

- **Bluetooth**: mainly used for computer input/output devices like keyboards and printers.
- **Ultra-wideband (UWB)**: mainly used for multimedia applications like video and audio delivery in home networks
- **Wi-Fi**: mainly used for internet and e-mails.
- **ZigBee**: mainly used for monitoring and controlling simple systems, like a passive infrared sensor system.

All these protocols are suitable for short-range communication, but they have different properties compared to each other. In table 2.6, properties can be seen for different wireless communication protocols [12] [14] [15].

Wireless communication technology	Bluetooth	UWB	Wi-Fi	ZigBee
IEEE standard	802.15.1	-	802.11 a/b/g	802.15.4
Frequency band	2.4 GHz	3.1 - 10.6 GHz	2.4 GHz; 5 GHz	868/915 MHz; 2.4 GHz
Max signal rate	1 Mb/s	110 Mb/s	54 Mb/s	250 kb/s
Nominal range	10 m	10 m	100 m	10 - 100 m
Nominal TX power	0 - 10 dBm	-41.3 dBm/MHz	15 - 20 dBm	(-25) - 0 dBm
Battery life	Several days	Several hours	Several days	Several years
Typical network join time	>3 sec	-	1 sec	30 ms
Protocol complexity	188 procedures	106 procedures	75 procedures	48 procedures

Table 2.6: Properties of wireless communication protocols.

By looking at table 2.6, it can be seen that the ZigBee protocol is the most suited protocol for the IPT application. The reason is because the ZigBee protocol has the lowest power consumption and the lowest complexity. The simplicity of the ZigBee protocol makes it suitable for the IPT application due to the limited memory use and needed computational capacity for the application [12]. The low maximum signal rate is not a problem, because the impedance matched set point doesn't need to be sent very often and this data is not very big. $U^*_{1,DC}$ is 32 bits long.

The chosen ZigBee module is the module XB24-AWI-001. This module has low power consumption and is easily programmable. Its specifications can be seen in Table 2.7.

XB24-AWI-001	
Range	90 m
Frequency	2.4 GHz
Transmit power	1 mW
Data rate	250 kb/s

Table 2.7: Specifications for XB24-AWI-001 [16].

2.4.3 UART system

The UART system consists of a couple of modules. On the transmitter side of the system, the modules that can be found are a buffer, a transmitter, a baud rate generator and a ZigBee module. The receiver side of the UART system has a ZigBee module, a receiver, a baud rate generator and a buffer. An overview of the UART system can be seen in figure 2.14.

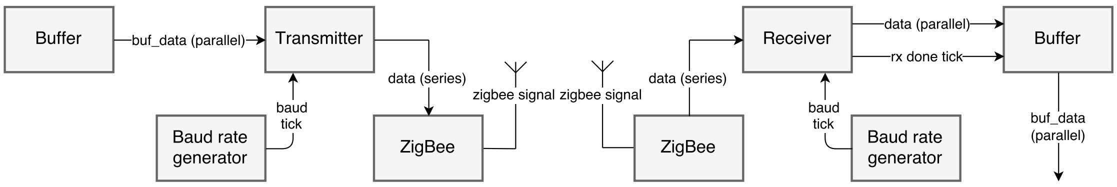


Figure 2.14: Overview of the UART system.

All the input and output signals of the modules are shown in the figure, except for the clock and reset signal. Of course, these signals are not omitted and are fed as input signals to every module in the UART system.

The buffer on the transmitter side will contain the value that needs to be transmitted through wireless communication to the receiver side. In the case of the IPT wireless charging system, this value is the impedance matched set point U^*_{1DC} with a length of 32 bits.
The baud rate generator has the job of telling the receiver and transmitter when a particular number of clock cycles has been repeated through the signal *baud tick*. This signal helps the transmitter and receiver with their tasks which will be explained later. This particular number of clock cycles is calculated with equation 2.16.

$$max\ counter = \frac{\text{clock frequency}}{\text{number of ticks in one bit} * \text{baud rate}} \tag{2.16}$$

The number of ticks in one bit is set to 16 ticks, which is a commonly used value [17]. A clock frequency of 50 MHz is used and the baud rate is set to 9600 bits per second, which is also commonly used [18]. After filling in equation 2.16, the maximum counter is:

$$max\ counter = \frac{50 * 10^6}{(16 * 9600)} = 326$$

The transmitter has the task of sending the value of $U^*_{1,DC}$ repeatedly to the receiver every 50 ms. The form of the transmitted data sequence can be seen in figure 2.15. The data sequence contains a start bit (bit value ’0’), 8 data bits and a stop bit (bit value ’1’). The start and stop bits are included so the receiver will know when a transmission starts and stops. The data sequence only contain 8 bits, because the ZigBee module can at most transmit 8 data bits at a time. So the transmitter has to transmit 4 times to get the value of $U^*_{1,DC}$ to the receiver. When the transmission is done, the transmitter will go in the idle state and wait until it needs to transmit again. During the idle state, the transmitter will continue transmitting the bit value ’1’. In order for the transmitter to know how long it needs to send each bit, it needs the input signal *baud tick*. After every 16 ticks from the baud rate generator, the transmitter will start to send a next bit.

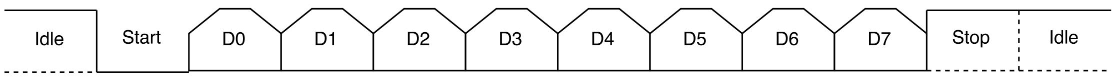


Figure 2.15: The transmitted data sequence through wireless communication.

The receiver has the job of storing the incoming data bits and putting them back into a 32 bits long sequence for the set point $U^*_{1,DC}$. After receiving the whole value of $U^*_{1,DC}$, the receiver will tell the buffer on the receiver side to store the set point value.
In the end, the buffer will contain the value of the impedance matched set point $U^*_{1,DC}$ that the control system on the primary side can use.

2.4.4 Error probability

From the Data sheet, it can be seen that the XB24-AWI-001 has a packet error rate of 1% for a signal strength of -92 dBm(6.3e-13W)[16]. The received signal power can be calculated using the Friis Formula [19]:

$$P_R = \frac{P_T G_T G_R \lambda}{4\pi R^2} \tag{2.17}$$

Where P_R is the received power, P_R is the transmitted power, G is the gain of the antennas, λ is the wavelength of the transmission and R is the distance between the transmitter and the receiver. From the datasheet:

$$P_T = 1mW \qquad G_T \approx 1.41 \qquad G_R \approx 1.41 \qquad \lambda = 0.125m$$

If the maximum distance between the 2 modules is taken to be 1m, filling in equation 2.17 gives a power of 0.02 mW. This is a factor 76 dB larger than the scenario for a 1% error rate. Because of this, the use of error detecting code was deemed unnecessary for errorless data transfer.

3 Evaluation

3.1 Overview

The main tool used to evaluate the design, will be an experimental prototype created using the selected components. Each component will be tested by measuring the output with a multimeter/oscilloscope to verify that the components work according to their specifications. The system will then be tested as a whole by performing a number of measurements over the entire measuring range. There will also be calculations done to verify the design requirements.

3.1.1 Requirements

Requirement	**Desired value**	**Method of evaluaton**
Current measurement resolution	0.01A	Comparing the results of practical measurements of the sensor readout over the entire measuring range to the measured value of a precision Multimeter.
Voltage sensor resolution	0.01V	Comparing the results of practical measurements of the sensor readout over the entire measuring range to the measured value of a precision Multimeter.
Wireless communication error rate	<1%	Calculated and tested using the experimental prototype

Table 3.1: An overview of the key system requirements and the methods used to verify that they are met.

3.2 Prototype

The implemented system prototype will perform the measurement of current and voltage, and wireless transmission simultaneously. This will be done by continuously acquiring data from both ADCs, processing them on the MCU, and transmitting them wirelessly over a distance of at least 1m. The receiving communication module will be connected to a computer and the received values will be displayed in a MATLAB program. Figure 3.1 shows a schematic of the prototype.

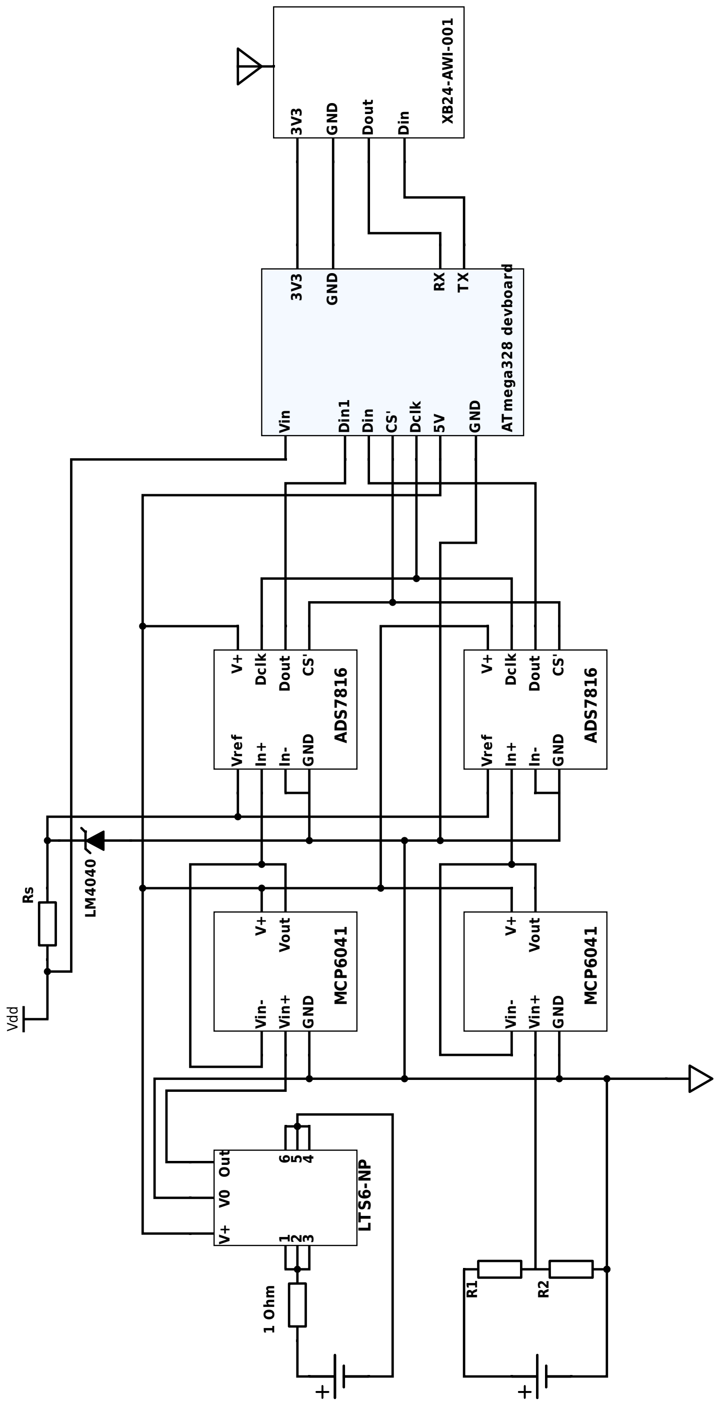


Figure 3.1: This figure shows the schematic for the experimental prototype.

The system consists of two 5 volt regulators. One on the control board, which will supply power to the components and one precision voltage reference, which is used as a stable reference for the ADCs. Both voltage regulators will be supplied with 6v from a lab bench power supply.

An overview of the resistor values and voltages used can be seen in table 3.2. a picture of the physical prototype can be seen in figure 3.2.

Table 3.2: an overview of the parameters used for the prototype.

parameter	value
R1	10.82 MΩ
R2	1 MΩ
Rs	3 kΩ
Vdd	6 V

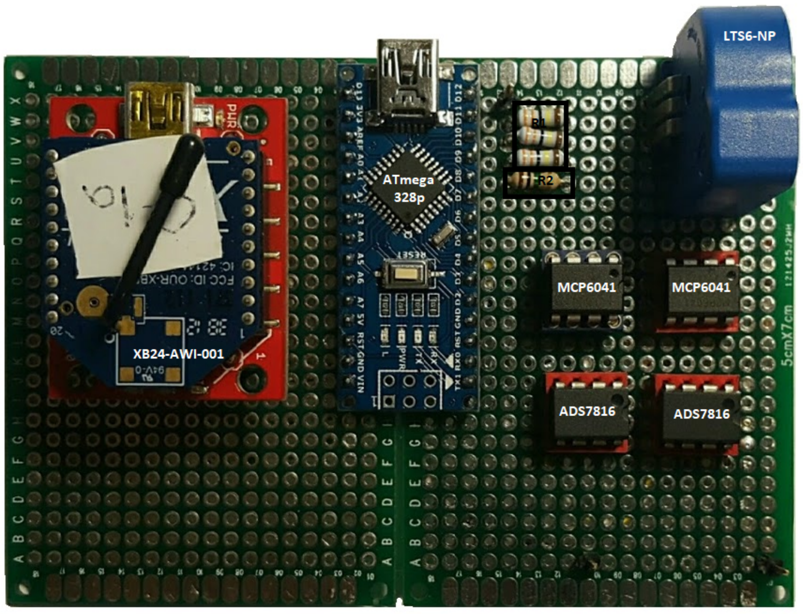


Figure 3.2: This figure shows a picture of the experimental prototype.

3.3 Testing and results

3.3.1 ADC

The ADC was tested by connecting a power supply to the input, transferring the output to the MCU, and printing the value to the serial port. The MCU is programmed using the Arduino IDE. Due to suspected noise from the power supply, the ADC error was measured to go up to 15 Least Significant Bits (LSB) over the range from 0 to 5 V. A smoothing function, which calculates the running average of 10 ADC samples was implemented on the MCU to filter noise from the data. After filtering, the error was verified to be 3 LSB at most.

3.3.2 Current measurement accuracy of 0.01A

The current sensor was first tested by connecting it to a 5V power supply. According to the datasheet, for a current of 0A the output value is equal to 2.5V. The sensitivity of the current sensor is 104.16mV/A. This was measured and verified using a precision multimeter. The offset for 0 A was within 4 mV.
After the current sensor was verified to be working, it was connected to the ADC through the buffer amplifier.
Using a Power supply and a 1 ohm power resistor, measurements were done over the range from 0 to 4A with increments of 0.5A. The results of the measurements were compared to the actual current. This can be seen in figure 3.3

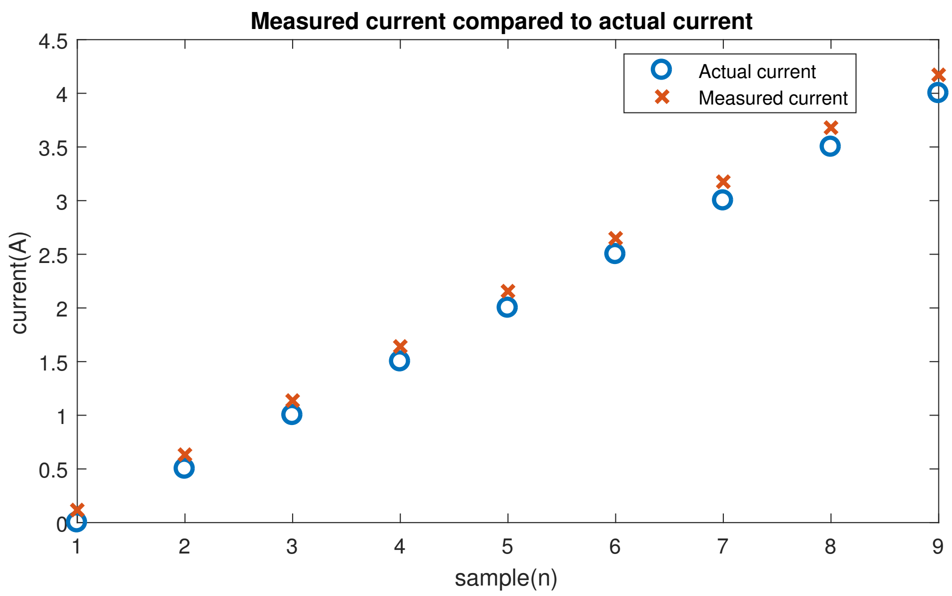


Figure 3.3: This graph shows the Measured current plotted against the actual current.

From the figure above, it may seem that the measurement error is constant over the entire measuring range. However, if the error is plotted separately as can be seen in figure 3.4, it can be seen that there is a small gain error. The error is roughly linear and can be approximated using the following first order polynomial:

$$E_{poly} = 0.0147 * I_{Measured} + 0.1196$$

Where E_{poly} is the approximated error, and $I_{Measured}$ is the measured current. The polynomial coefficients are calculated using MATLAB.

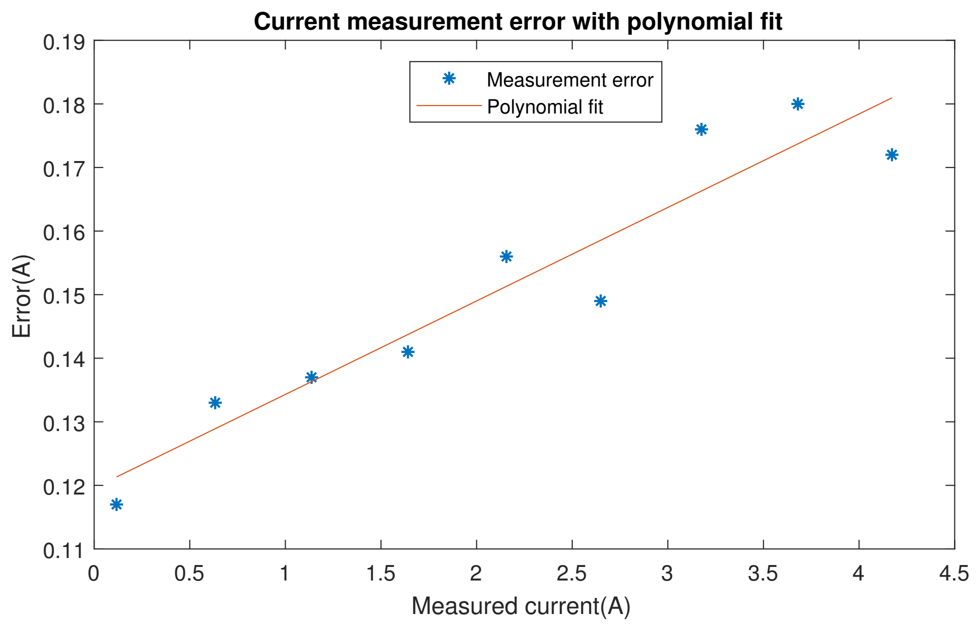


Figure 3.4: This figure shows the current measurement error and a polynomial fit through the data.

Using the MCU, this polynomial can be added to the initial measured current as a bias. Performing the measurements for a second time with the added bias yields the result seen in figure 3.5. The measurement error after biasing can be seen in figure 3.6

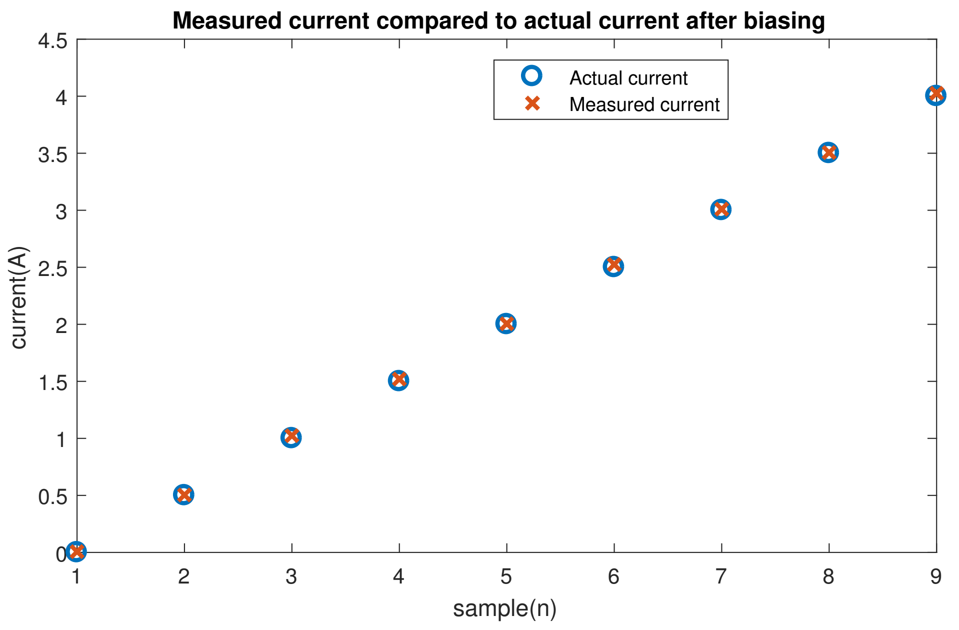


Figure 3.5: This figure shows the biased measured current compared to the actual current.

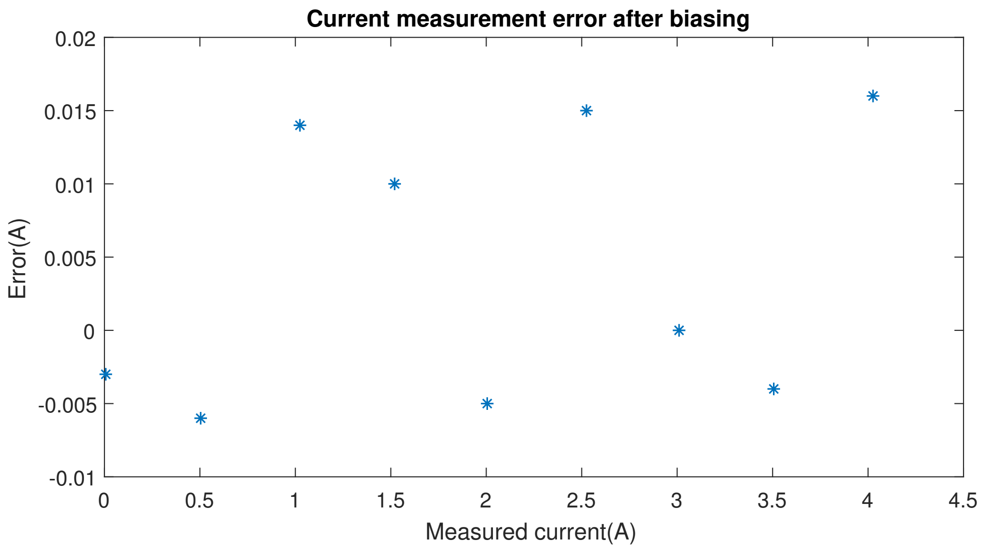


Figure 3.6: This figure shows the current measurement error after biasing.

The maximum error after biasing is equal to 0.0160. The average error is equal to 0.0041 A. This is within the system requirements.

3.3.3 Voltage measurement accuracy of 0.1V

The voltage ratio of the voltage divider can be calculated with the following equation:

$$R = \frac{R_1}{R_1 + R_2} \tag{3.1}$$

Where R is the divider ratio and R_1 and R_2 are the values of the resistors as shown in figure 3.1. Filling in the equation using table 3.2 gives a value of 0.0924 for the prototype. Considering that 5 V is the maximum input voltage of the ADC, the maximum voltage that can be measured is given by,

$$V_{range} = \frac{5}{0.0924} = 54.11V$$

The voltage divider is connected to the ADC through the buffer amplifier as seen in figure 3.1. The output of the ADC is processed using the MCU and the value is printed to the USB port. Since the power supply used had a maximum output voltage of 20 V, the measuring range was limited. The voltage was measured with increments of 1V. The result can be seen in figure 3.7

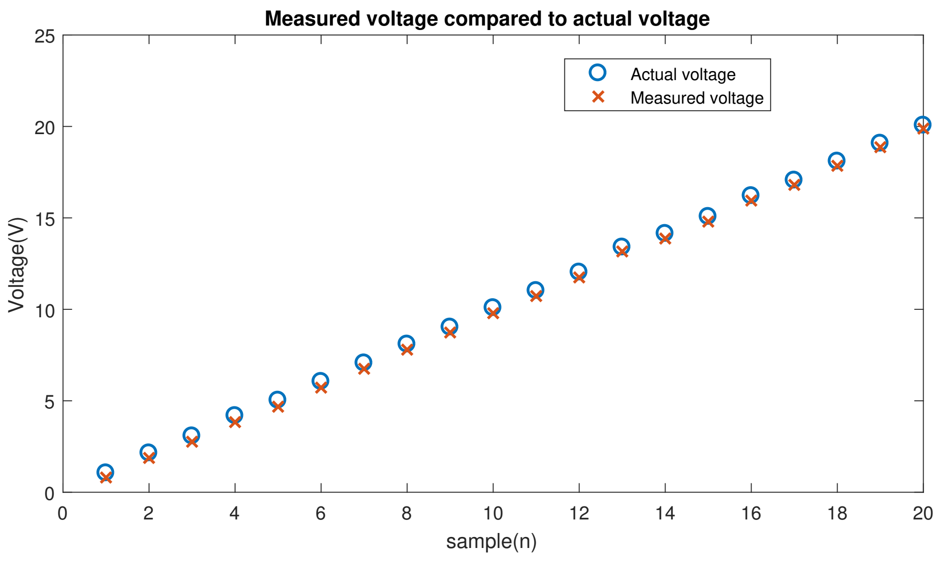


Figure 3.7: This figure shows the measured voltage compared to the actual voltage measured by a multimeter.

Similarly to the current measurements, it can be seen from figure 3.8 that there is a small offset and gain error. The error can be approximated by,

$$E_{poly} = -0.0050 * V_{measured} + 0.3289$$

Where $V_{measured}$ is initial measured voltage.

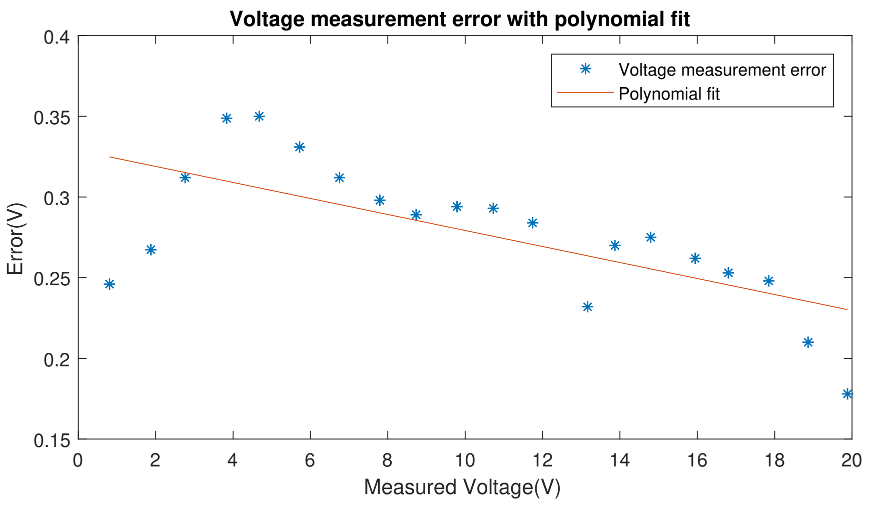


Figure 3.8: This figure shows the voltage measurement error and a polynomial fit through the data.

After biasing the measured voltage in the MCU, the measurements were redone. The results can be seen in figure 3.9. The error after biasing is shown in figure 3.10

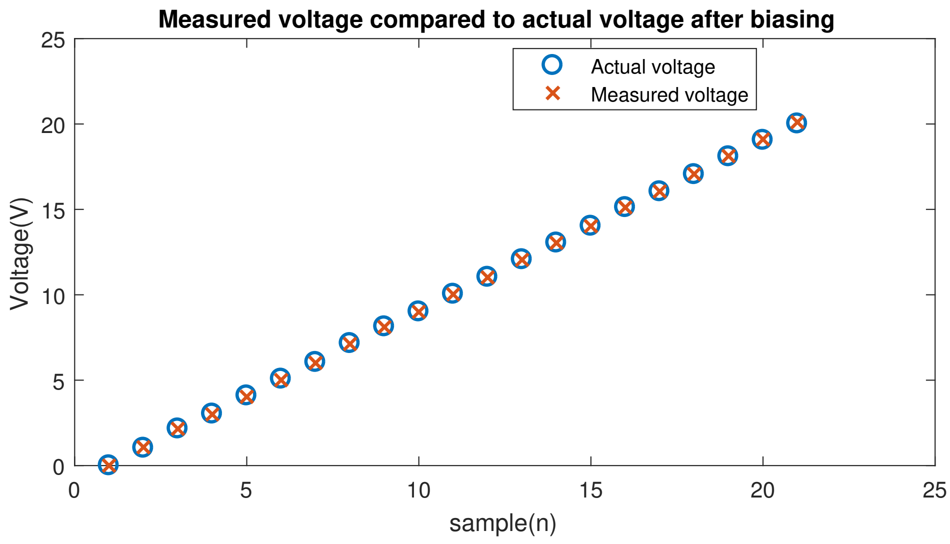


Figure 3.9: This figure shows the voltage measurement after biasing compared to the actual voltage.

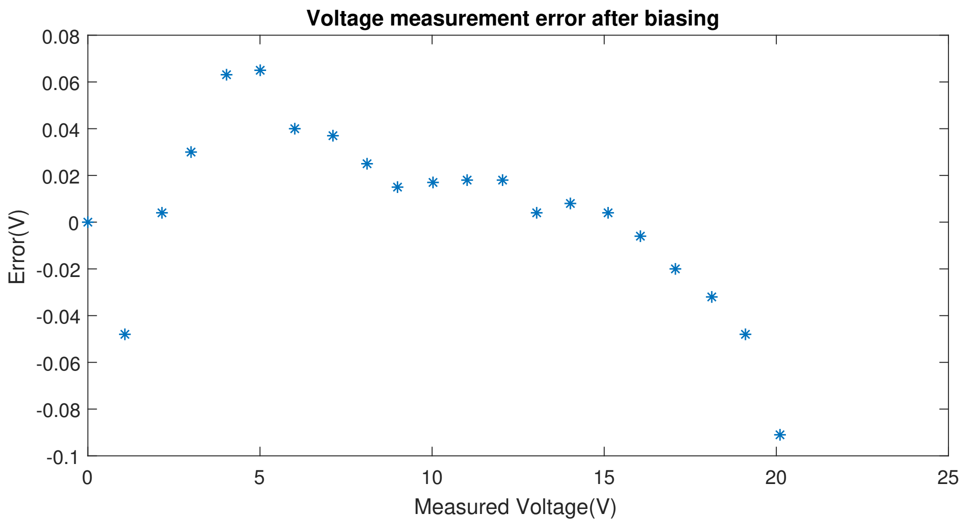


Figure 3.10: This figure shows the voltage measurement error after biasing.

The maximum measurement error after biasing is equal to 0.09V and the average value of the error is equal to 0.0049 V. This is within the system requirements.

3.3.4 Wireless communication with error rate <1%

In order for two Xbee modules to be able to communicate, they must be programmed with the correct destination, and source addresses. This is done using XCTU. The xbee modules are also configured with no parity. To transmit the measured values wirelessly, the transmitting xbee module is connected to the UART of the MCU. The arduino IDE has a built in library of functions which can be used to transmit the data sequentially to the MCUs UART. The code can be seen in Appendix A.3. In order to test the wireless communication, the measured values for the current and voltage will be wirelessly sent to a receiver module. The receiving xbee module will be connected to a computer via USB. In order to read out the incoming sensor values, a matlab app is created. The UI can be seen in figure 3.11. The code can be seen in appendix A.4.

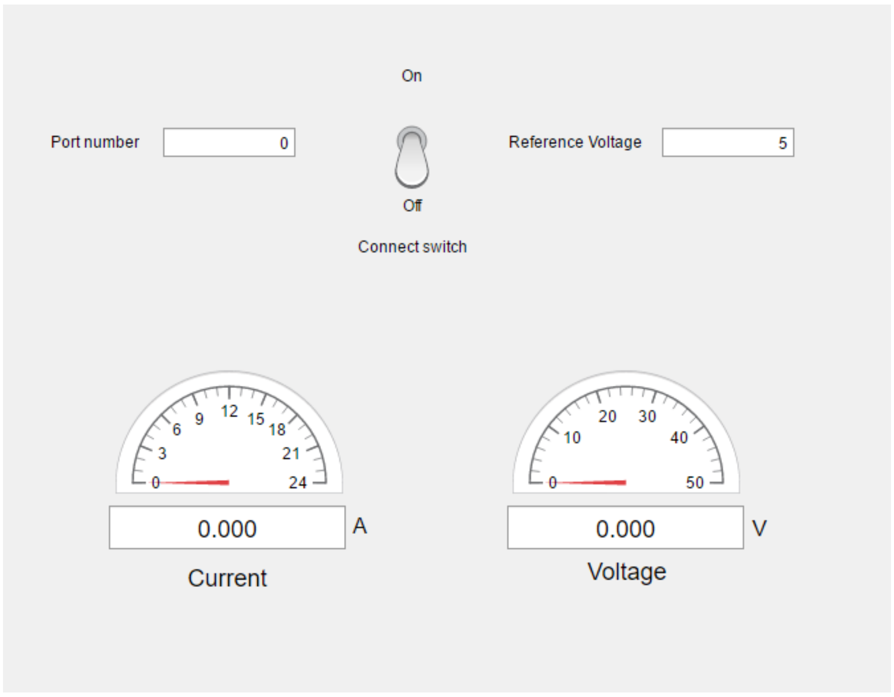


Figure 3.11: This figure shows the user interface of the program used to read the wirelessly transmitted sensor data.

The program works by first filling in the COM port number of the receiving xbee module and turning the connect switch on. There is also a voltage reference adjust field in order to adjust the reference voltage if needed. After the program is started, data from the comport is continuously read and displayed in the corresponding text box and gauge. From the xbee datasheet the error rate is equal to 1% for a distance of 100m. For the required distance of 1m, it is safe to assume that the error rate is within the system requirements. During testing, no incorrect packets were received.

4 Conclusion

4.1 Assessment

In conclusion, this report discussed the design and testing prototype of a wireless communication and sensor system. The experimental prototype was able to measure current and voltage with an accuracy of around 0.2%. The system is fairly modular, meaning that the range of voltages and currents that can be measured can easily be increased or decreased by changing the current sensor and the resistors for the voltage divider. The system is also able to wirelessly communicate with a receiver module without error. The power use of the system is fairly low, as the entire system uses only 0.6W of power. The system does, however, have some limitations. The system is only able to measure positive DC voltages. This is within the requirements of the system, but makes it less flexible for different applications. Also, the circuit used for voltage measurement requires a significant amount of calibration since the resistors used for the voltage divider have a fairly high resistance tolerance. This can be improved by using high precision resistors.

4.2 Next steps

The next steps would be to create a prototype specific to the design. This prototype will need to have an interface to accept user input for a desired charging power. The system must then calculate the set points for both control systems and relay this value to the control systems. Depending on how the control systems are implemented, an interface needs to be created to relay sensor data and set-point values to the control system. For this prototype, the measurement accuracy can also be improved by adding low pass LC filters to the input and supply of the ADCs to filter out noise. The power use of the system can be further improved by implementing a sleep mode for the MCU and the communication module when wireless transmission or sensor data is not required.

Bibliography

[1] "2016 global plug-in light vehicle sales increased by about 80% in 2015 | department of energy." https://energy.gov/eere/vehicles/fact-918-march-28-2016-global-plug-light-vehicle-sales-increased-about-80-2015. (Accessed on 05/31/2017).

[2] S. D. Brown and Z. G. Vranesic, *Fundamentals of digital logic with VHDL design.* McGraw-Hill, 2008.

[3] I. Kuon, J. Rose, and R. Tessier, *FPGA architecture: survey and challenges.* Now, 2008.

[4] "Atmega328/p." http://www.atmel.com/Images/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf. (Accessed on 07/29/2017).

[5] M. Pelgrom, *Analog to digital conversion*, p. 249–318. Springer International Publishing, 2017.

[6] "12-bit high speed micro power sampling analog-to-digital converter." http://www.ti.com/lit/ds/sbas061/sbas061.pdf. (Accessed on 07/29/2017).

[7] "Mcp6041/mcp6042/mcp6043/mcp6044 data sheet." http://www.dexsilicium.com/Microchip_MCP6041.pdf. (Accessed on 07/29/2017).

[8] S. Ziegler, R. C. Woodward, H. H.-C. Iu, and L. J. Borle, "Current sensing techniques: A review," *IEEE Sensors Journal*, vol. 9, no. 4, p. 354–376, 2009.

[9] "Current transducer lts 6-np datasheet." http://www.lem.com/docs/products/lts_6-np.pdf. (Accessed on 06/02/2017).

[10] B. Ferreira and W. van der Merwe, *The Principles of Electronic and Electromechanic Power Conversion.* John Wiley and Sons Inc, 2014.

[11] O. Knecht and J. W. Kolar, "Comparative evaluation of ipt resonant circuit topologies for wireless power supplies of implantable mechanical circulatory support systems," pp. 3271–3278, March 2017.

[12] J. S. Lee, Y. W. Su, and C. C. Shen, "A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wi-fi," pp. 46–51, Nov 2007.

[13] I. Lita, S. Oprea, I. B. Cioc, and D. A. Visan, "Wireless technologies for distributed sensor networks used in measurement and automation systems," pp. 303–307, May 2008.

[14] T. Liu, J. Liu, and B. Liu, "Design of intelligent warehouse measure and control system based on zigbee wsn," *2010 IEEE International Conference on Mechatronics and Automation*, 2010.

[15] A. Asaduzzaman, K. K. Chidella, and M. F. Mridha, "A time and energy efficient parking system using zigbee communication protocol," pp. 1–5, April 2015.

[16] "Xbee-datasheet.pdf." https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-Datasheet.pdf. (Accessed on 06/18/2017).

[17] P. P. Chu, *FPGA Prototyping by VHDL Examples, Xilinx Spartan-3 version.* John Wiley and Sons Inc, 2008.

[18] "basys2:basys2_rm.pdf." https://reference.digilentinc.com/_media/basys2:basys2_rm.pdf. (Accessed on 06/15/2017).

[19] L. W. Couch, *Digital and Analog Communication Systems.* Pearson Education, 2013.

Appendix

A.1 Calculations for the load resistor

$$Z_p = j\omega L_p + \frac{1}{j\omega C_p} = j\omega(L_1 - M) - j\frac{\omega^2 L_1}{\omega} = j\omega(L_1 - M) - j\omega L_1 = -j\omega M$$

$$Z_s = j\omega L_s + \frac{1}{j\omega C_s} = j\omega(L_2 - M) - j\frac{\omega^2 L_2}{\omega} = j\omega(L_2 - M) - j\omega L_2 = -j\omega M$$

$$\begin{aligned} Z_{total} &= R_p + Z_p + (Z_s + R_s + R_{load}) \parallel j\omega M = R_p - j\omega M + \frac{j\omega M(-j\omega M + R_s + R_{load})}{-j\omega M + R_s + R_{load} + j\omega M} \\ &= R_p - j\omega M + \frac{\omega^2 M^2 + j\omega M(R_s + R_{load})}{R_s + R_{load}} = R_p + \frac{\omega^2 M^2}{R_s + R_{load}} \end{aligned}$$

$$I_{in} = \frac{U_{1,AC}}{Z_{total}} = \frac{U_{1,AC}}{R_p + \frac{\omega^2 M^2}{R_s + R_{load}}} = \frac{U_{1,AC}(R_s + R_{load})}{R_p(R_s + R_{load}) + \omega^2 M^2}$$

$$\begin{aligned} I_{load} &= \frac{j\omega M I_{in}}{j\omega M + Z_s + R_s + R_{load}} = \frac{j\omega M}{R_s + R_{load}} \frac{U_{1,AC}(R_s + R_{load})}{R_p(R_s + R_{load}) + \omega^2 M^2} \\ &= \frac{j\omega M U_{1,AC}}{R_p(R_s + R_{load}) + \omega^2 M^2} \end{aligned}$$

$$P_{load} = \frac{\omega^2 M^2 U_{1,AC}^2 R_{load}}{\left[R_p(R_s + R_{load}) + \omega^2 M^2\right]^2}$$

$$P_{in} = \frac{U_{1,AC}^2}{R_p + \frac{\omega^2 M^2}{R_s + R_{load}}} = \frac{(R_s + R_{load}) U_{1,AC}^2}{R_p(R_s + R_{load}) + \omega^2 M^2}$$

Power transfer efficiency:

$$\begin{aligned} \eta = \frac{P_{load}}{P_{in}} &= \frac{\omega^2 M^2 U_{1,AC}^2 R_{load}}{\left[R_p(R_s + R_{load}) + \omega^2 M^2\right]^2} \frac{R_p(R_s + R_{load}) + \omega^2 M^2}{(R_s + R_{load}) U_{1,AC}^2} \\ &= \frac{\omega^2 M^2 R_{load}}{R_p(R_s + R_{load})^2 + \omega^2 M^2 (R_s + R_{load})} = \frac{\omega^2 M^2 R_{load}}{R_p(R_s^2 + 2R_s R_{load} + R_{load}^2) + \omega^2 M^2 (R_s + R_{load})} \end{aligned}$$

$$\frac{d\eta}{dR_{load}} = \frac{\left[R_p(R_s+R_{load})^2+\omega^2M^2(R_s+R_{load})\right]\omega^2M^2-\omega^2M^2R_{load}(2R_pR_s+2R_pR_{load}+\omega^2M^2)}{\left[R_p(R_s+R_{load})^2+\omega^2M^2(R_s+R_{load})\right]^2}$$

By setting the derivative equation for the efficiency equal to 0, MATLAB then gives:

$$\begin{aligned}R_{load} &= \frac{1}{R_p}\sqrt{R_pR_s(M^2\omega^2+R_pR_s)} = \frac{1}{R_p}\sqrt{R_pR_sM^2\omega^2+R_p^2R_s^2} = \sqrt{M^2\omega^2\frac{R_s}{R_p}+R_s^2}\\ &\approx \sqrt{M^2\omega^2\frac{R_s}{R_p}} = M\omega\sqrt{\frac{R_s}{R_p}}\end{aligned}$$

A.2 Calculations of the inverter voltage output

Periodic signals can be described by equation 1 which shows the quadrature form of the Fourier series. The variables c_k and d_k are the Fourier coefficients, and they are described by 2 and 3, respectively [19].

$$x(t) = c_0 + 2\sum_{k=1}^{\infty}\left[c_k\cos(k\omega t)+d_k\sin(k\omega t)\right] \tag{1}$$

$$c_k = \frac{1}{T}\int_{-T/2}^{T/2} x(t)\cos(k\omega t)\,\mathrm{d}t, \qquad k = 0,1,2,... \tag{2}$$

$$d_k = \frac{1}{T}\int_{-T/2}^{T/2} x(t)\sin(k\omega t)\,\mathrm{d}t, \qquad k = 1,2,3,... \tag{3}$$

For the Fourier series of $U_{1,DC}$, d_k is equal to 0. This is because the used square wave is an even function and the wave cannot be constructed by odd functions. So only the coefficients

c_k have values, and the calculations for c_k are described by the following equations:

$$\begin{aligned}
c_k &= \frac{1}{T}\int_{-T/2}^{T/2} x(t)\cos(k\omega t)\,\mathrm{d}t \\
&= \frac{U_{1,DC}}{T}\int_{-\frac{\alpha T}{4\pi}}^{\frac{\alpha T}{4\pi}} \cos(k\omega t)\,\mathrm{d}t - \frac{U_{1,DC}}{T}\int_{\frac{T}{2}-\frac{\alpha T}{4\pi}}^{\frac{T}{2}} \cos(k\omega t)\,\mathrm{d}t - \frac{U_{1,DC}}{T}\int_{-\frac{T}{2}}^{-\frac{T}{2}+\frac{\alpha T}{4\pi}} \cos(k\omega t)\,\mathrm{d}t \\
&= \frac{U_{1,DC}}{T}\frac{T}{2\pi k}\left\{\left[\sin(k\omega t)\right]_{-\frac{\alpha T}{4\pi}}^{\frac{\alpha T}{4\pi}} - \left[\sin(k\omega t)\right]_{\frac{T}{2}-\frac{\alpha T}{4\pi}}^{\frac{T}{2}} - \left[\sin(k\omega t)\right]_{-\frac{T}{2}}^{-\frac{T}{2}+\frac{\alpha T}{4\pi}}\right\} \\
&= \frac{U_{1,DC}}{2\pi k}\left\{\left[\sin\left(k\frac{\alpha}{2}\right)+\sin\left(k\frac{\alpha}{2}\right)\right] - \left[\sin(k\pi) - \sin\left(k\pi - k\frac{\alpha}{2}\right)\right] - \left[-\sin\left(k\pi - k\frac{\alpha}{2}\right)+\sin(k\pi)\right]\right\} \\
&= \frac{U_{1,DC}}{\pi k}\left[\sin\left(k\frac{\alpha}{2}\right) - \sin(k\pi) + \sin\left(k\pi - k\frac{\alpha}{2}\right)\right] \\
&= \frac{U_{1,DC}}{\pi}\left[\frac{\sin\left(k\frac{\alpha}{2}\right)}{k} - \frac{\sin(k\pi)}{k} + \frac{\sin\left(k\pi - k\frac{\alpha}{2}\right)}{k}\right]
\end{aligned}$$

$$\begin{aligned}
c_0 &= \frac{U_{1,DC}}{\pi}\lim_{k\to 0}\left[\frac{\sin\left(k\frac{\alpha}{2}\right)}{k} - \frac{\sin(k\pi)}{k} + \frac{\sin\left(k\pi - k\frac{\alpha}{2}\right)}{k}\right] \\
&= \frac{U_{1,DC}}{\pi}\lim_{k\to 0}\left[\frac{\alpha}{2}\cos\left(k\frac{\alpha}{2}\right) - \pi\cos(k\pi) + \left(\pi - \frac{\alpha}{2}\right)\cos\left(k\pi - k\frac{\alpha}{2}\right)\right] \\
&= \frac{U_{1,DC}}{\pi}\left(\frac{\alpha}{2} - \pi + \pi - \frac{\alpha}{2}\right) = 0
\end{aligned}$$

$$\begin{aligned}
c_{1,3,5,\ldots} &= \frac{U_{1,DC}}{\pi}\left[\frac{\sin\left(k\frac{\alpha}{2}\right)}{k} - \frac{\sin(k\pi)}{k} + \frac{\sin\left(k\pi - k\frac{\alpha}{2}\right)}{k}\right] \\
&= \frac{U_{1,DC}}{\pi}\left[\frac{\sin\left(k\frac{\alpha}{2}\right)}{k} + \frac{\sin\left(\pi - k\frac{\alpha}{2}\right)}{k}\right] = \frac{U_{1,DC}}{\pi}\left[\frac{\sin\left(k\frac{\alpha}{2}\right)}{k} + \frac{\sin\left(k\frac{\alpha}{2}\right)}{k}\right] \\
&= \frac{2U_{1,DC}}{\pi}\frac{\sin\left(k\frac{\alpha}{2}\right)}{k}
\end{aligned}$$

$$\begin{aligned}
c_{2,4,6,\ldots} &= \frac{U_{1,DC}}{\pi}\left[\frac{\sin\left(k\frac{\alpha}{2}\right)}{k} - \frac{\sin(k\pi)}{k} + \frac{\sin\left(k\pi - k\frac{\alpha}{2}\right)}{k}\right] \\
&= \frac{U_{1,DC}}{\pi}\left[\frac{\sin\left(k\frac{\alpha}{2}\right)}{k} + \frac{\sin\left(2\pi - k\frac{\alpha}{2}\right)}{k}\right] = \frac{U_{1,DC}}{\pi}\left[\frac{\sin\left(k\frac{\alpha}{2}\right)}{k} - \frac{\sin\left(k\frac{\alpha}{2}\right)}{k}\right] \\
&= 0
\end{aligned}$$

So the Fourier coefficients c_k are:

$$c_k = \begin{cases} 0 & \text{for } k = 0 \\ \frac{2U_{1,DC}}{\pi} \frac{\sin\left(k\frac{\alpha}{2}\right)}{k} & \text{for } k = 1,3,5,... \\ 0 & \text{for } k = 2,4,6,... \end{cases} \tag{4}$$

By substituting the values of equation 4 in the Fourier series equation, then the equation for the inverter voltage output is:

$$\begin{aligned} U_{1,AC} &= c_0 + 2\sum_{k=1}^{\infty} \left[c_k \cos(k\omega t) + d_k \sin(k\omega t)\right] \\ &= 2 \sum_{k=1,3,5,...}^{\infty} \frac{2U_{1,DC}}{\pi} \frac{\sin\left(k\frac{\alpha}{2}\right)}{k} \cos(k\omega t) = \frac{4}{\pi} U_{1,DC} \sum_{k=1,3,5,...}^{\infty} \frac{\sin\left(k\frac{\alpha}{2}\right)}{k} \cos(k\omega t) \end{aligned} \tag{5}$$

A.3 MCU code

```
#define CLOCK 2    // clock pin
#define DATAPIN 3  // data pin for current measurement
#define CS 4       // chip select data pin for adc
#define DATAPIN2 5 // data pin for voltage measurement



int currentint;
int voltageint;
float measured_current;
float measured_voltage;
float voltagebias;


//parameters initialization

const int numReadings = 10;          //number of readings used to filter the input
float vol_ref = 5.0;        // reference voltage for the adc        (voltage)
float vol_div_ratio = 0.0924;   // ratio of the voltage divider           (ratio)
float cursensitivity = 0.10416;    // sensitivity of the current sensor      (volt/ampe
//float voltagebias = 0.277;       // accounting for adc offset          (voltage)
//float currentbias = 0;      // accounting for adc offset          (amperage)



int readings_cur[numReadings];
int readings_vol[numReadings];

int readIndex = 0;                  // the index of the current reading

int total_cur = 0;                  // the running total
int total_vol = 0;                  // the running total

int average_cur = 0;                // the average filtered value for the current
int average_vol = 0;                // the average average filtered value for the volta


void setup()
{
  //start serial communicaton with xbee and select pin modes for all pins
  Serial.begin(9600);
  pinMode(CS, OUTPUT);
  pinMode(DATAPIN, INPUT);
  pinMode(CLOCK, OUTPUT);
  digitalWrite(CS, HIGH);
  digitalWrite(CLOCK, LOW);

// set all elements in the array to 0 to initialize the average value
for (int thisReading = 0; thisReading < numReadings; thisReading++) {
    readings_cur[thisReading] = 0;
    readings_vol[thisReading] = 0;
  }

}


void loop(){
  currentint = 0;
  voltageint = 0;

  digitalWrite(CS, LOW);          //begins adc conversion
  clockchange ();
  clockchange ();                 //data is output after two clock cycles
  for (int i = 11; i >= 0; i--)
  {
    clockchange ();
    currentint+=digitalRead(DATAPIN)<<i;
    voltageint+=digitalRead(DATAPIN2)<<i;
  }

  digitalWrite(CS, HIGH);        // stops adc conversion

  total_cur = total_cur - readings_cur[readIndex];
  total_vol = total_vol - readings_vol[readIndex];

  readings_cur[readIndex]  = currentint;
  readings_vol[readIndex]  = voltageint;

  total_cur = total_cur + readings_cur[readIndex];
  total_vol = total_vol + readings_vol[readIndex];

  readIndex = readIndex + 1;



  if (readIndex >= numReadings) {
    readIndex = 0;
  }

  // calculate the average:
  average_cur = total_cur / numReadings;
  average_vol = total_vol / numReadings;


  // convert to actual measurements

  measured_current=((((float(average_cur)/4096.0)*vol_ref)-2.5)/cursensitivity);
float current_bias = 0.0896+0.0147*measured_current;
float  measured_current_biased=measured_current-current_bias;


 measured_voltage=(((float(average_vol)/4096.0)*vol_ref)/vol_div_ratio);
if (measured_voltage == 0)
{
  voltagebias=0;
}
else
{
  voltagebias=0.3567;
}
float measured_voltage_biased=measured_voltage+voltagebias;

  // send data to xbee
  Serial. print("current: ");
  Serial.print(measured_current_biased,3);
  Serial.print("\t voltage: ");
  Serial.print(measured_voltage_biased,3);
  Serial.println();
  delay(50);
}
```

```
// function to perform clock cycle
void clockchange (void)
{
  digitalWrite(CLOCK, HIGH);
  digitalWrite(CLOCK, LOW);
}
```

A.4 Matlab UI code

```
classdef app2 < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                        matlab.ui.Figure
        PortnumberLabel                 matlab.ui.control.Label
        portnumber                      matlab.ui.control.NumericEditField
        VoltageEditFieldLabel           matlab.ui.control.Label
        VoltageEditField                matlab.ui.control.NumericEditField
        VLabel                          matlab.ui.control.Label
        ALabel                          matlab.ui.control.Label
        CurrentEditFieldLabel           matlab.ui.control.Label
        CurrentEditField                matlab.ui.control.NumericEditField
        ConnectswitchSwitchLabel        matlab.ui.control.Label
        ConnectswitchSwitch             matlab.ui.control.ToggleSwitch
        Curgauge                        matlab.ui.control.SemicircularGauge
        Volgauge                        matlab.ui.control.SemicircularGauge
        ReferenceVoltageEditFieldLabel  matlab.ui.control.Label
        ReferenceVoltageEditField       matlab.ui.control.NumericEditField
    end


    methods (Access = private)


    end


    methods (Access = private)

        % Value changed function: ConnectswitchSwitch
        function ConnectswitchSwitchValueChanged(app, event)
        values = app.ConnectswitchSwitch.Value;
        values= pad(values,3);

        if values == 'On '
         comport = strcat('COM',int2str(app.portnumber.Value));
         xbee = serial(comport);
         set(xbee,'BaudRate',9600);
         fclose(xbee);
         fclose(instrfind);
         fopen(xbee);
         while(1)
         sensvalstr=fscanf(xbee);
         sensvalstr=pad(sensvalstr,35);

         currentvalstr=sensvalstr(strfind(sensvalstr,'current')+9 ...
         :strfind(sensvalstr,'voltage')-1);
         voltagevalstr=sensvalstr(strfind(sensvalstr,'voltage')+9 ...
         :strfind(sensvalstr,'voltage')+15);
         drawnow;

         currentval=str2double(currentvalstr);
         if isnan(currentval)
             currentval=0;
         end
         currentvalog=(currentval*0.10416+2.5)/5*4096;
         currentval=(currentvalog/4096*app.ReferenceVoltageEditField.Value-2.5) ...
         /0.10416;

         drawnow;
         voltageval=str2double(voltagevalstr);
         if isnan(voltageval)
             voltageval=0;
         end
         %voltageval=40;
         voltagevalog=(voltageval*0.0924)/5*4096;
         voltageval=(voltagevalog/4096*app.ReferenceVoltageEditField.Value) ...
         /0.0924;
         drawnow;

         app.CurrentEditField.Value = currentval;
         app.Curgauge.Value =abs(app.CurrentEditField.Value);

```

```
            app.VoltageEditField.Value = voltageval;
            app.Volgauge.Value =abs(voltageval);




                if values == 'Off'

                    break;

                end
                drawnow;
            end
            fclose(instrfind);
           end

               1;


        end
    end

    % App initialization and construction
    methods (Access = private)

        % Create UIFigure and components
        function createComponents(app)

            % Create UIFigure
            app.UIFigure = uifigure;
            app.UIFigure.Position = [100 100 675 522];
            app.UIFigure.Name = 'UI Figure';

            % Create PortnumberLabel
            app.PortnumberLabel = uilabel(app.UIFigure);
            app.PortnumberLabel.HorizontalAlignment = 'right';
            app.PortnumberLabel.Position = [35 409 72 15];
            app.PortnumberLabel.Text = 'Port number';

            % Create portnumber
            app.portnumber = uieditfield(app.UIFigure, 'numeric');
            app.portnumber.Position = [125 405 100 22];

            % Create VoltageEditFieldLabel
            app.VoltageEditFieldLabel = uilabel(app.UIFigure);
            app.VoltageEditFieldLabel.HorizontalAlignment = 'center';
            app.VoltageEditFieldLabel.FontSize = 18;
            app.VoltageEditFieldLabel.Position = [442.5 81 66 23];
            app.VoltageEditFieldLabel.Text = 'Voltage';

            % Create VoltageEditField
            app.VoltageEditField = uieditfield(app.UIFigure, 'numeric');
            app.VoltageEditField.ValueDisplayFormat = '%.3f';
            app.VoltageEditField.HorizontalAlignment = 'center';
            app.VoltageEditField.FontSize = 18;
            app.VoltageEditField.Position = [385 109 179 33];

            % Create VLabel
            app.VLabel = uilabel(app.UIFigure);
            app.VLabel.HorizontalAlignment = 'center';
            app.VLabel.FontSize = 16;
            app.VLabel.Position = [563 115 25 20];
            app.VLabel.Text = 'V';

            % Create ALabel
            app.ALabel = uilabel(app.UIFigure);
            app.ALabel.HorizontalAlignment = 'center';
            app.ALabel.FontSize = 16;
            app.ALabel.Position = [261 117 25 20];
            app.ALabel.Text = 'A';

            % Create CurrentEditFieldLabel
            app.CurrentEditFieldLabel = uilabel(app.UIFigure);
            app.CurrentEditFieldLabel.HorizontalAlignment = 'center';
            app.CurrentEditFieldLabel.FontSize = 18;
            app.CurrentEditFieldLabel.Position = [140.5 76 66 23];
            app.CurrentEditFieldLabel.Text = 'Current';

            % Create CurrentEditField
            app.CurrentEditField = uieditfield(app.UIFigure, 'numeric');
            app.CurrentEditField.ValueDisplayFormat = '%.3f';
            app.CurrentEditField.HorizontalAlignment = 'center';
            app.CurrentEditField.FontSize = 18;
            app.CurrentEditField.Position = [84 109 179 33];

            % Create ConnectswitchSwitchLabel
            app.ConnectswitchSwitchLabel = uilabel(app.UIFigure);
            app.ConnectswitchSwitchLabel.HorizontalAlignment = 'center';
            app.ConnectswitchSwitchLabel.Position = [269.5 330 88 15];
            app.ConnectswitchSwitchLabel.Text = 'Connect switch';

            % Create ConnectswitchSwitch
            app.ConnectswitchSwitch = uiswitch(app.UIFigure, 'toggle');
            app.ConnectswitchSwitch.ValueChangedFcn = createCallbackFcn(app, @Connectsw
            app.ConnectswitchSwitch.Position = [297 381 32 72];

            % Create Curgauge
            app.Curgauge = uigauge(app.UIFigure, 'semicircular');
            app.Curgauge.Limits = [0 24];
            app.Curgauge.Position = [89 151 172 93];

            % Create Volgauge
            app.Volgauge = uigauge(app.UIFigure, 'semicircular');
            app.Volgauge.Limits = [0 50];
            app.Volgauge.Position = [389 151 172 93];

            % Create ReferenceVoltageEditFieldLabel
            app.ReferenceVoltageEditFieldLabel = uilabel(app.UIFigure);
            app.ReferenceVoltageEditFieldLabel.HorizontalAlignment = 'right';
            app.ReferenceVoltageEditFieldLabel.Position = [380 409 107 15];
            app.ReferenceVoltageEditFieldLabel.Text = 'Reference Voltage';

            % Create ReferenceVoltageEditField
            app.ReferenceVoltageEditField = uieditfield(app.UIFigure, 'numeric');
            app.ReferenceVoltageEditField.ValueDisplayFormat = '%11.5g';
            app.ReferenceVoltageEditField.Position = [502 405 100 22];
            app.ReferenceVoltageEditField.Value = 5;
        end
    end

    methods (Access = public)

        % Construct app
        function app = app2

            % Create and configure components
            createComponents(app)

            % Register the app with App Designer
            registerApp(app, app.UIFigure)

            if nargout == 0
                clear app
            end
        end

        % Code that executes before app deletion
        function delete(app)

            % Delete UIFigure when app is deleted
            delete(app.UIFigure)
        end
    end
end
```