# INTRODUCTION

## Objectives

1. To propose a blockchain-enabled P2P market model of generated PV energy.
2. To propose a dual power source controller for switching power between the internal SHS battery and external grid connection.

# METHODOLOGY

## The Virtual Layer Model

This section proposes a model to be used for setting up a blockchain network and a protocol for negotiating on the assets to be traded leading up to a successful transaction.

The Multichain blockchain platform was selected as the platform upon which the P2P blockchain network proposed in this paper. Multichain is an open source blockchain platform that enables users to create and deploy private blockchains. The platform was chosen for this study for the following reason:

1. The platform offers private blockchains This allows the generated blockchain to be only visible to the peers trading energy on the microgrid. This is advantageous in that it allows control of the transactions over the network. In addition to this, the costly proof of work mechanism integral to other open blockchains such a Bitcoin can be avoided.
2. The platform is developer friendly. An API and command-line interface (CLI) are provided to users to interact with their blockchain. This platform is advantageous in that it reduces development time considerably via its API and CLI interfaces; and also enable a developer to custom-make their blockchain via creating the tradable assets on the network and specifying their native trading currencies.

Three integral parties are involved in this simulation: prosumers have energy generating capabilities and sell their excess energy to consumers, consumers have no PV generation capacity and buy their energy preferentially from prosumers or from a central utility, and the network admin initializes the network, adding peers and issuing crypto currency coins for transactions.

The following pseudo-code demonstrates the steps involved in initializing the blockchain for initial use:

1. Begin
2. Create Blockchain
3. Create energy & e-coin assets
4. Create Prosumer\_Capacities open stream
5. Create Consumer\_Demands open stream
6. Create Peer nodes.
7. Grant peers connect, send, and receive permissions
8. For each P2P connection
9. Create private stream
10. End for
11. End

The energy asset represents a unit of electricity in kWh that is to be traded over blockchain. The e-coin asset is the native currency used to trade assets over blockchain.

Streams provide a natural abstraction for blockchain use cases which focus on general data retrieval, timestamping and archiving, rather than the transfer of assets between participants (Multichain, 2020). Streams are used to implement a key-value database in this methodology. The “prosumer-capacities” stream is used by prosumers to publish their available excess energy. The “consumer-demands” stream is used by consumers to publish their energy demand. Consumers can view the published capacities of prosumers at a given moment via the “Prosumer\_Capacities” stream and prosumers view consumer demands via the “Consumer\_Demands” stream. These streams mimic the physical market place better where a buyer and seller are able to publish their demand and supply.

A key is used to identify each item published on the streams. The JSON object used to publish consumer demands or prosumer’s available excess energy is structured as follows:

{ “amount” : *amount* },

Where amount is the consumer electricity demands in kWh prosumer excess energy for supply in kWh.

Private streams are created for each pair of nodes. Using these streams, a consumer and prosumer are able to negotiate for a settlement, upon which a transaction is made for the agreed amount. This mechanism solves the problem of partial transaction experienced by Kang, et al. (2017). This is demonstrated later in the Results section.

JSON objects are used in these streams to facilitate information exchange. The following picture demonstrates a simple protocol developed for information exchange, leading up to a successful trade. For this demonstration, a prosumer initiates the negotiation, but the converse also applies.

|  |  |  |
| --- | --- | --- |
| Publish electricity demand  Receive proposal | public  Consumer-Demands | Read Consumer-Demands  Propose energy supply |

prosumer

consumer

Figure .: Initiating P2P trade proposal on blockchain

The JSON object sent by a prosumer to show interest in supplying energy to a consumer with public demand on the public Consumer-Demands stream is structured as shown:

{ “interested”:true , “amount\_available”:*amount* }

Using this, the prosumer notifies the consumer if he/she has exactly the amount of energy requested by the consumer, or can only cater for part of the request. If the consumer agrees to the prosumer proposition, the transaction is made complete. This is described by the diagram below:

|  |  |  |
| --- | --- | --- |
| Commit to proposal  Receive energy  Send e-coin asset |  | Receive commit instruction  Send energy asset  Receive e-coin |

consumer

prosumer

Figure .: Trading P2P assets on blockchain

An exchange rate of 1 e-coin for 1 kWh of electricity is assumed in this model as used by NRGcoin (NRGcoin, 2020). The P2P energy trade instance is then complete.

## The Physical Layer Model

### Prosumer Section

Requirements Analysis

The following requirements dictate the demands the proposed system had to satisfy:

* The system must simulate a SHS setup.
* The system must simulate electricity input from the grid during undercharged periods.
* The system must simulate electricity export during overcharged periods.

Design

The following block diagram below of the proposed system is shown below.

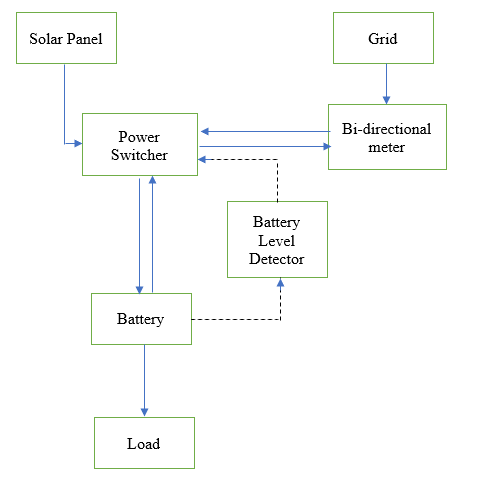


Figure .: Prosumer Control system block diagram

The load is primarily powered by the energy stored in the battery. The system may receive power from the local solar panel, present in the SHS, another prosumer via the grid, or central utility via the grid. A power switcher circuit is at the heart of this design. This circuit determines from where the power feeding the system is to be sourced. The decision from where the power switcher receives power is enabled by the battery level detector.

The following three scenarios dictate the operation of this system:

1. If the battery is undercharged, the power switcher sources power from the grid.
2. If the battery is level is within operating range, power is sourced from the solar panels.
3. If the battery is overcharged, excess power is exported to other peers via the grid.

### Consumer Section

Requirements Analysis

The following demands dictate the requirements the consumer system must satisfy:

* The system must simulate electricity input from the grid.

# IMPLEMENTATION AND RESULTS

## Simulating P2P trade between peers in Uttara

Initializing blockchain network

This section presents the results obtained from implementing the procedure proposed in section 3.2.

A new blockchain named “uttarap2pblockchain” was created by the admin.

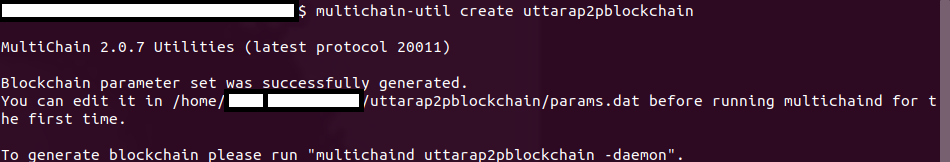


Figure .: Creating new blockchain

The blockchain was then started for the first time. A genesis block was as so created.

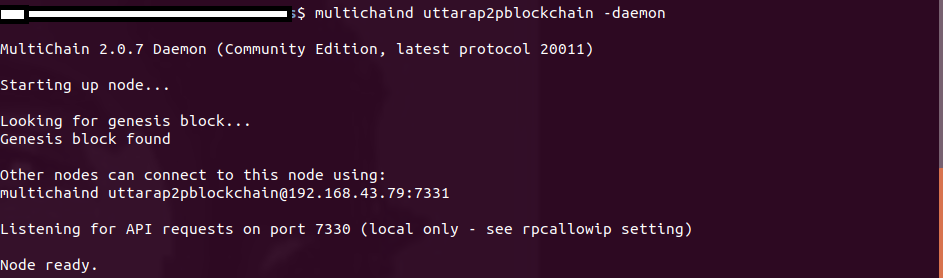


Figure . : Starting new blockchain

The energy asset was created:

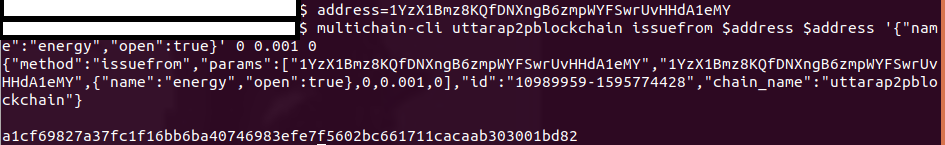


Figure .: Creating energy asset

The e-coin asset was similarly successfully.created

The public “consumer-demands” stream to be used by consumers to publish their energy demnds was then created.

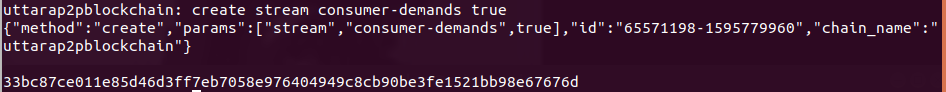
****

Figure .: Creating "consumer-demands" stream

The “prosumer-capacities” stream was similarly successfully created.

A prosumer node was created on the localhost,

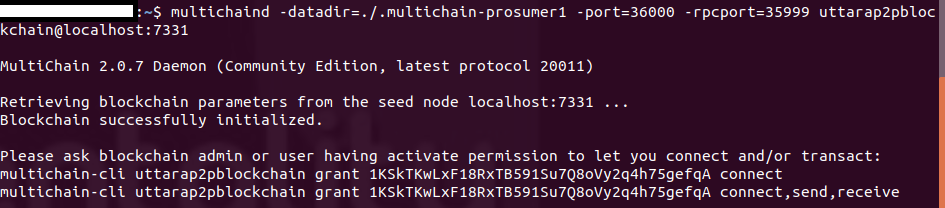


Figure .: Creating prosumer node

and granted permission by the admin to connect to, send and receive on the blockchain.

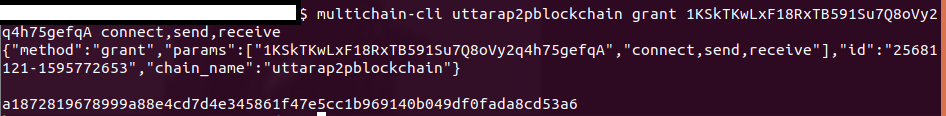


Figure .: Granting prosumer connect, send, receive permissions

This node was then started successfully.

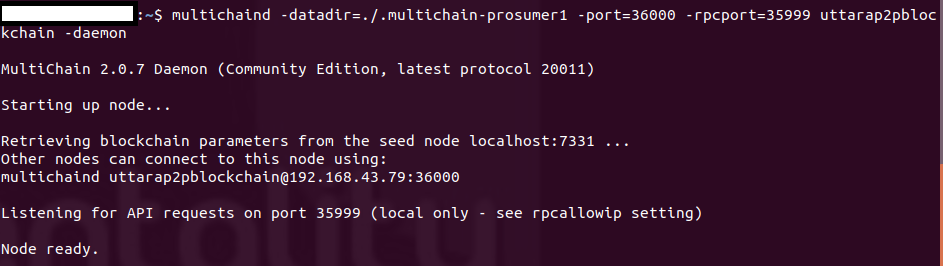


Figure .: Starting prosumer node

A consumer node was similarly created, granted permissions by the admin, and started.

The consumer then proceeded to publish an open demand for 3kWh of electricity.

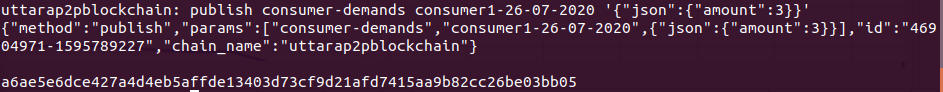
****

Figure .: Consumer publishing energy demand

The prosumer received the demand,

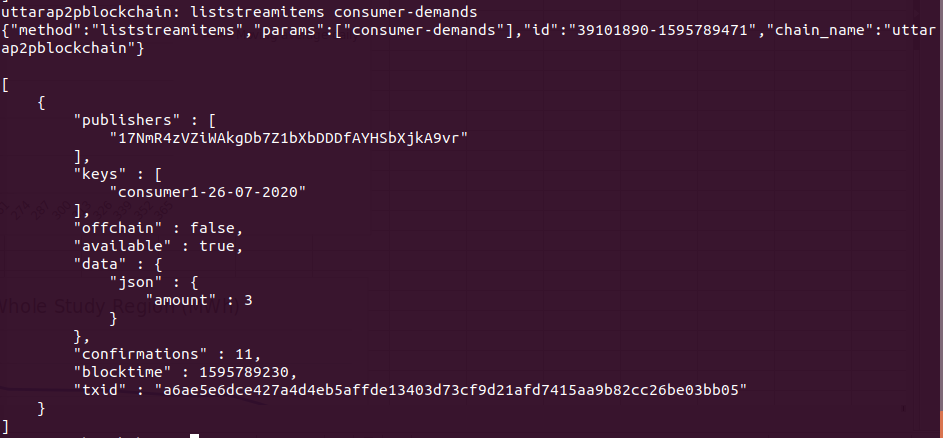
****

Figure .: Prosumer viewing consumer energy demands

**a**nd expressed interest in fully supplying to the consumer, using a private stream dedicated for the particular pair only.

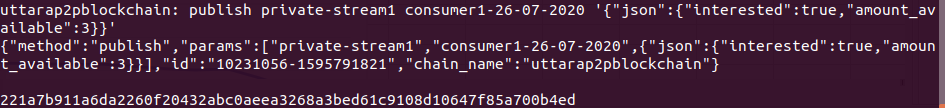
****

Figure .: Prosumer expressing interest in supplying energy

The consumer then received this proposal via their private stream,

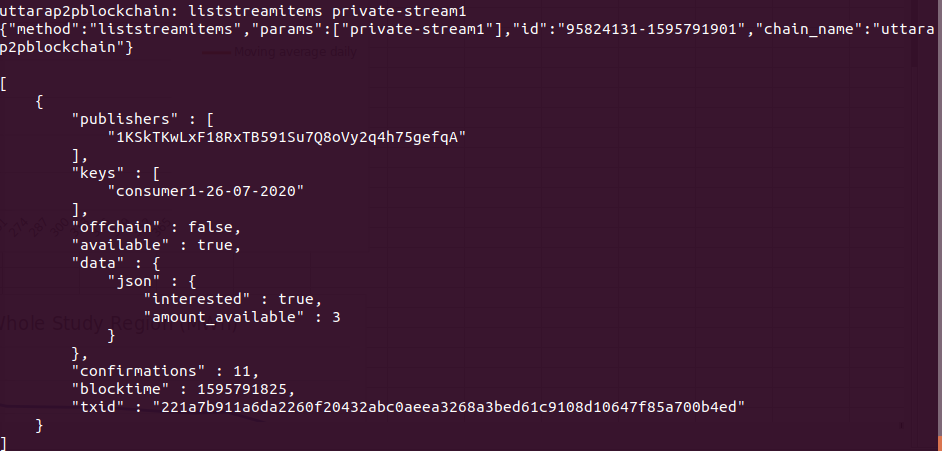
****

Figure .: Consumer receiving prosumer proposal

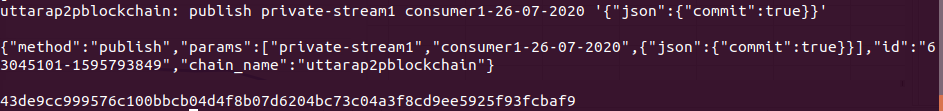
and agreed to commit to the proposal.****

Figure .: Consumer committing to prosumer's proposal

The pair then proceeded to exchange the prosumer’s excess energy for the consumer’s e-coin as follows. The prosumer sent the consumer his/her energy asset.

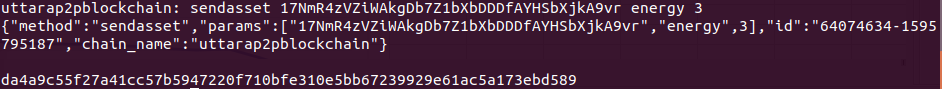
****

Figure .: Prosumer sending energy asset

The consumer then received the energy asset, and subsequently sent his/her e-coin asset.



Figure .:Consumer receiving energy asset

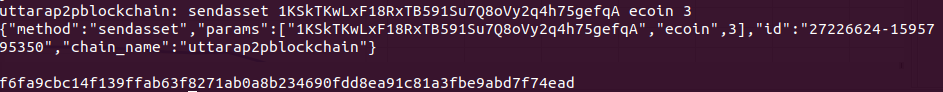
****

Figure .: Consumer sending e-coin asset

The prosumer then verified receival of the e-coin asset into his/her wallet.

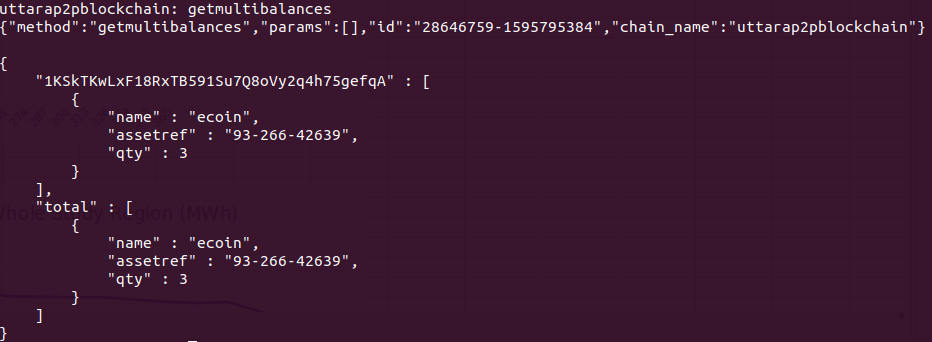
****

Figure .: Prosumer verify e-coin receival

The transaction was then complete. At the end of this transaction, the prosumer had exchanged 3KWh of eleectrcity for 3 e-coin, while the consumer exchanged 3 e-coin for3 KWh of electricity.

## Control Systems

### Prosumer

The block diagram designed in Figure 3.5 was synthesized, and the flowing schematic obtained as such.

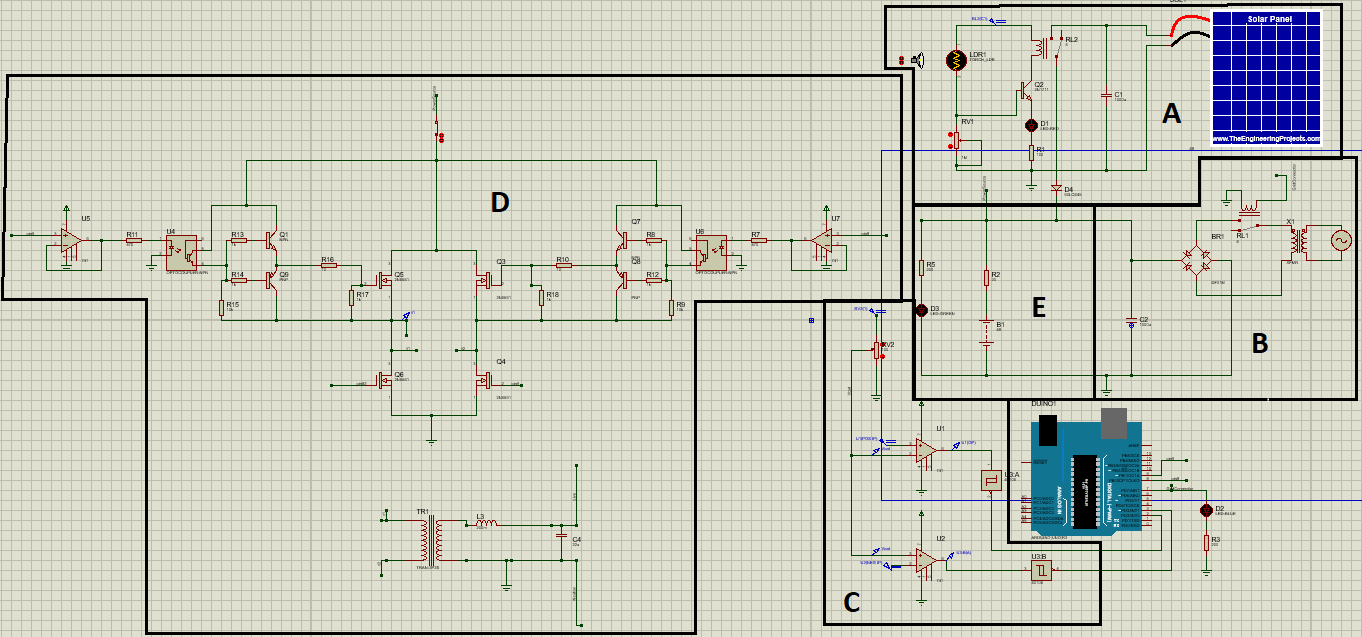


Figure .: Prosumer schematic

The schematic can be divided into five section:

**Section A** : This section models the solar production section. The solar panel was modeled in this schematic was adapted from Nasir (2018). A light dependent resistor (LDR) circuit controls the connection of the solar panel power into the load. During high sunlight hours, the voltage divider betwork comprised of the light dependent resistor LDR1 and variable resistor RV1 forward biases the npn transistor Q2, allowing a current to flow through the collector, drain terminals of the resistor, thus magnetizing the relay RL2. This allowed the solar panel to be connected to the load, charging the battery, and feeding the load. A sample of this in action was captured and displayed in Figure 4.18 below. The sensitivity of the circuit is varied by the variable resistor RV1.

**Section B :** This is a full wave bridge rectifier circuit. This section rectifies AC power incoming from the grid via the diode bridge circuit. The capacitor C2 smoothens the rectified power into a constant level. Relay RL1 controls the activation of this section. The relay is triggered by the control logic provided by the Arduino. The bridge connection is normally open. A sample screenshot of the Arduino switching on the bridge circuit in the case scenario of undercharged battery is demonstrated in Figure 4.19

**Section C :** This section simulates the battery level detection section. Two operational amplifiers are employed in this section as comparators. The first comparator provided by op amp U1 detects when the 48V battery is overcharged. When the battery is overcharged, marking surplus energy availablity, the prosumer is capable of selling his/her excess energy.

The second comparator provided by op amp U2 detects when the battery is under charged. When the battery is undercharged, marking a period of under production, the prosumer is necessitated to import power from the grid connection.

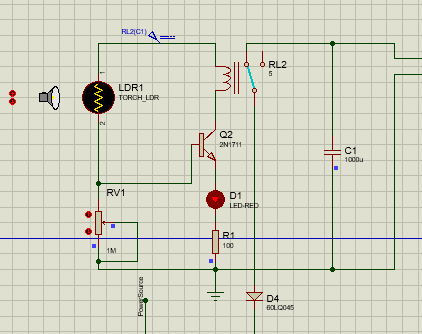


Figure .: Solar Panel Connected

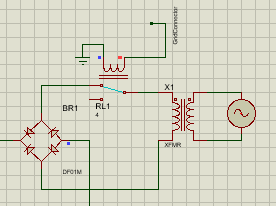


Figure .: Rectifier circuit triggered on by Arduino

This section works in conjuction with the Arduino board, employing the two interrupt pins to detect an overcharged or undercharged scenario. Two Schmitt triggers were used to produce sharp rissing or falling edges for the Arduino interrupt pins to detect.

The following diagram shows the transitions imposed by the comparators to the Arduino interrupt pins during different test scenarios.

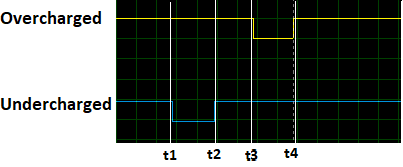


Figure .: Battery level detector operation

It was observed that the comparator U1 tracking the overcharged condition went from high to low when the battery was overcharged (t3), stayed low during this overcharged condition (t3-t4), and went from low to high when the battery was discharged below overcharged voltaged (t4).

On the other hand, the comparator U2 tracking the undercharged condition went from high to low when the battery was undercharged (t1), remained low during this condition (t1-t2), and wenr highwhen the battery was charged above the undercharged voltage (t2).

**Section D :** This is an the inverter section. An inverter was necessitated to convert battery DC power to AC power, in order to be exported to the grid and sold to other peers on the microgrid. The basic premise behind the circuit is the H-bridge circuit, composed of n-channel enhancement mosfets, as redacted in Figure 4.20. N-channel mosfets are turned on by positive gate voltages. In order for a the mosfet to be completely turned on, it is necessitated that the gate termnal potential be higher than the source terminal potential (Tahmid, 2013).

This condition that the gate terminal potential must be higher than the source terminal voltage presents a challenge for the high-side mosfets, Q5 and Q3. As such, the Arduino 5V potential couldn’t directly turn on these high-side mosfets, as opposed to the low-side mosftes Q6 and Q4. To fix this, a drive circuit was provided for these mosfets, as discussed by Tahmid (2013), with operational amplifiers providing isolation of the Arduino pins from the driver sciruits.

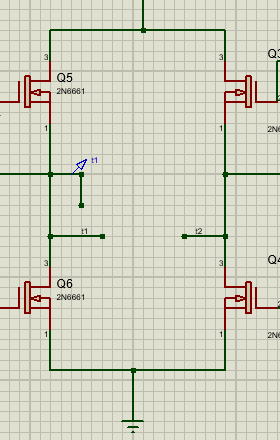


Figure .: H-bridge section

The Arduino board produces positive trigger voltages using pin9 and pin8 to activate the mosfets as deemed necessary to produce an AC signal. The H-bridge circuit producess a 50Hz square-shaped AC waveform between point t1 and t2 as captured by the oscilloscope output below.

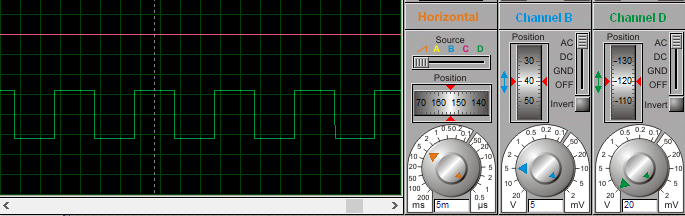


Figure .: Square-wave from H-bridge

This signal was coupled to an a resonant LC network to carve the square-wave to a sine wave via the transformer TR1. The transformer was chosen to provide isolation of the square-wave supply from the LC network; and such prevent loading effectes presented upon the souce by the network.

An oscilloscope output of the sine wave produced is shown below.

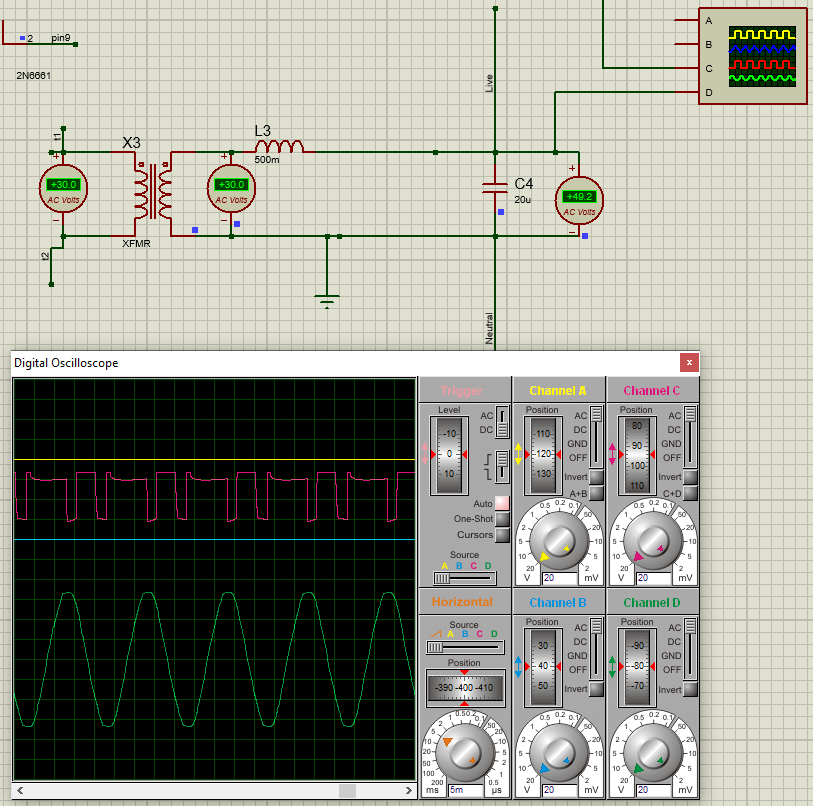


Figure .: Square wave shaped to sine wave

It was , observed that despite the inclusion of the transformer, the original square-wave was distorted. However, the sine wave produced retained the 50Hz frequency of the original ac square-wave.

**Section E :** This section simulates a simple 200Ω, in series with a green LED.

The Arduino UNO board was the controller for this design. Two interrupt pins, pin2 and pin3 were used to detect an overcharged or undercharged battery consition. Two digital pins, pin9 and pin8 were used to provide trigerring signals for the inverter gates. The code executed by the microcontroller is attached in the Appendices section.

### Consumer

The schematic diagram below displays the results of implementing the consumer system.

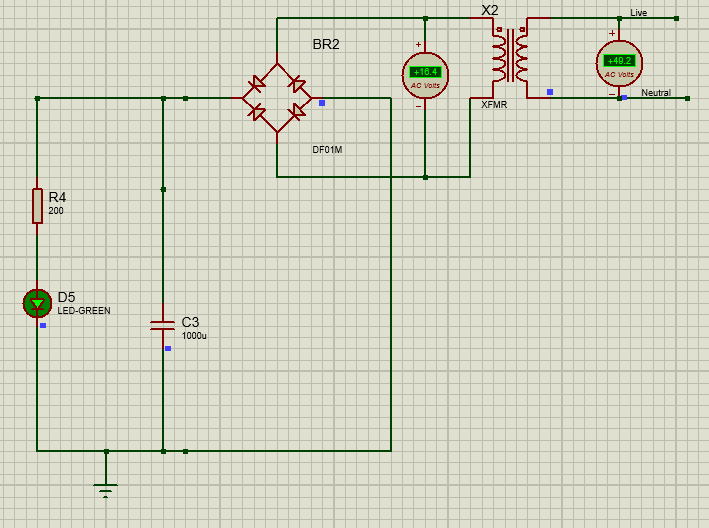


Figure .: Consumer Model

The ac power exported to the grid by the prosumer was imported by the consumer. A transformer with a turns ratio 3 was used in this model. This was value was chosen to reduce the grid 220V down to 73V, which the diodes comprising the rectifier circuit can tolerate. For his simulation, however, the desired 220V was not achieved.

# DISCUSSION

## The Virtual Layer

The virtual layer was simulated using the Multichain platform. The proposed model allowed a prosumer to propose to the consumer the amount of energy he/she was capable of selling to him/her, after the consumer published an open request for a specific amount of energy demand. It was inferred that the proposed model was capable of solving the issue of partial transactions faced by Kim, et al. (2017), by allowing smaller exchenges to be conducted after information exchange using private streams.

## The Physical Layer

The following observations were made for the proposed simulation model for the prosumer, synthesized and implemented in Figure4.17.

* Solar power was able to be simulated. The LDR circuit developed enabled the solar power to be connected and disconnected from charging the battery via a relay.
* Grid power was able to be switched on and off, depending on the battery level. However, the design was not able to authenticate this procedure depending on whether or not the prosumer had paid for this external power.
* An inverter design enabled the prosumer to sell his/her excess energy to a consumer. This design, however, was not able to integrate with the negotiation demonstrated in the virtual layer. A simulation of stepping up output power from the inverter circuit to 220V as necessitated by the national grid was also not feasibly demonstrated. An output RMS power of 50V was at 50Hz was achieved in this simulation.
* A battery level detection circuit was demonstrated, and integrated with the Arduino UNO control system. The detection of the battery level enabled the prosumer to switch on the circuitry controlling input of the external grid power into the local environment. This process, would naturally be a precursor to the prosumer placing a demand to receive external power.

# Appendices

Arduino Code

const int pin1 = 9;

const int pin2 = 8;

volatile int output = LOW;

volatile bool changed = false;

void setup() {

// put your setup code here, to run once:

attachInterrupt(digitalPinToInterrupt(2),overcharged, FALLING);

attachInterrupt(digitalPinToInterrupt(3), undercharged, FALLING);

pinMode(pin1, OUTPUT);

pinMode(pin2, OUTPUT);

pinMode(7, OUTPUT);

digitalWrite(7, LOW);

}

void loop() {

digitalWrite(pin1, HIGH);

digitalWrite(pin2, LOW);

delay(10);

digitalWrite(pin1, LOW);

digitalWrite(pin2, HIGH);

delay(10);

digitalWrite(pin2, LOW);

}

void overcharged(){

output = LOW;

digitalWrite(7,output);

}

void undercharged(){

output = HIGH;

digitalWrite(7,output);

}

nline]. Available: http://pwr.company/.

[Accessed: May-2017]

”pwr.company,” pwr.company. [Online]. Available: http://pwr.company/.

[Accessed: May-2017]

[3]” pwr.company,” pwr.company. [Online]. Available: http://pwr.company/.

[Accessed: May-2017]