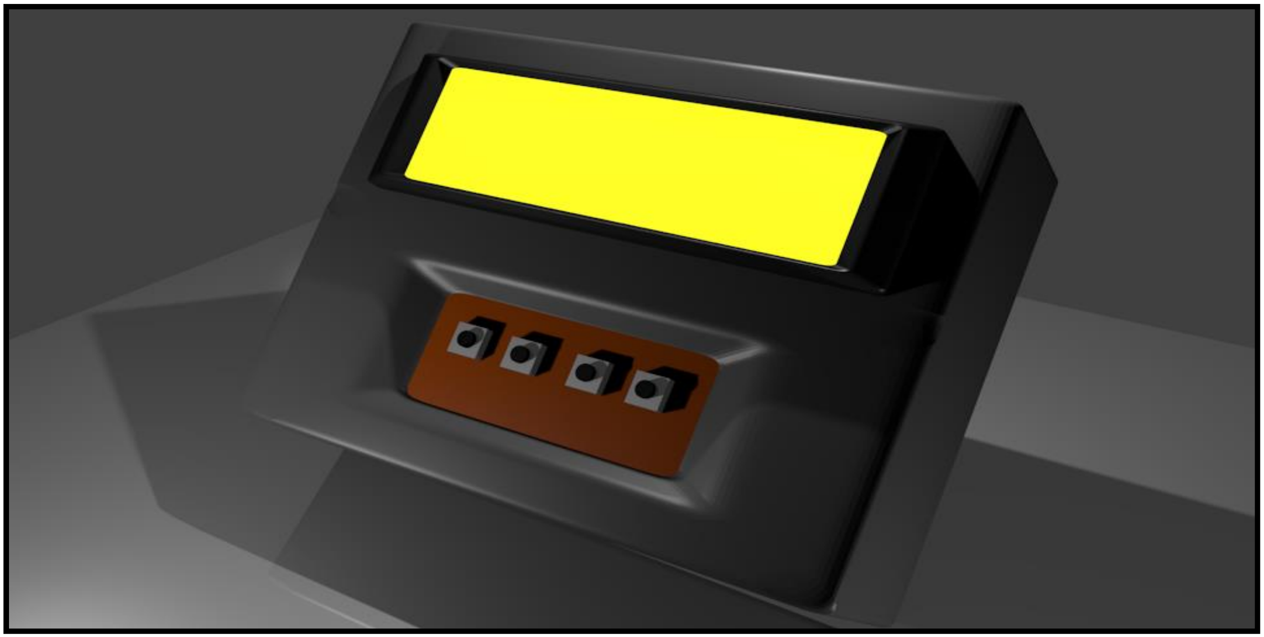
AC Fluctuation

Timer and Counter



AC Fluctuation Timer & Counter is a device that detects and counts the number of fluctuation occurred. The device measures its duration and it can accurately save the time in its memory when a fluctuation eventuated.

The timer and counter feature of the device is triggered by a Power Relay, connected to the I/O Pin of Gizduino ATmega328P microcontroller unit. The device can measure and save data up to 30 fluctuation occurrence. All data and measurements are streamed through LCD screen (16x2). Other features within the device are Time and Date Display and Settings (Backlight – ON/OFF & Sleep Timer).

Features

- Fluctuation Timer & Counter
- Total Fluctuation Display
- Time & Date Display
- Settings

General Specifications

Gizduino ATmega328P

- Power Input: 8 – 12V_{DC}
  USB: 5V_{DC}
- Clock Speed: 16MHz
  12MHz for programming

OMRON MY-2NJ 220V_{AC}

- 2PDT
- Coil Voltage: 220/240 V_{AC}
- Max Contact Voltage:
  - 125V_{DC}
  - 250V_{AC}

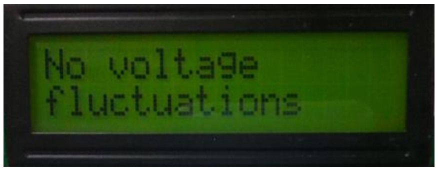
For detailed specifications, please refer to Gizduino and Omron MY Relay manual

Table of Contents

I. Features

I.1. Fluctuation Timer and Counter

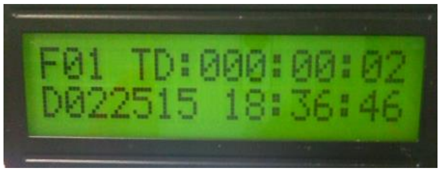
Displays the time, date and duration of the current fluctuation occurred.



No voltage
fluctuations

Figure I.1.1

Figure I.1.1 is displayed when there are no current fluctuations have occurred.



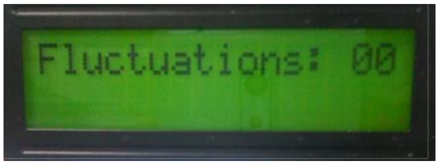
F01 TD:000:00:02
D022515 18:36:46

Figure I.1.2

Figure I.1.2 is the display format for every fluctuation events. It displays the following data:

F01 → Current fluctuation count (from 01 to 30)
D022515 → Date when a fluctuation occurred
First 2-Digits → Month (02)
3rd and 4th Digits → Day (25)
Last 2-Digits → Year (Current year - 2000)
18:36:46 → Time when a fluctuation occurred (24-Hour Format)
TD → Current fluctuation duration (hours-minutes-seconds, total hours in 3-Digit Format)

I.2. Total Fluctuation Display

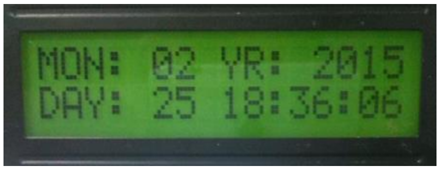


Fluctuations: 00

Figure I.2.1

Figure 1.2.1 is the display format for total number of fluctuations.

I.3. System Time and Date



MON: 02 YR: 2015
DAY: 25 18:36:06

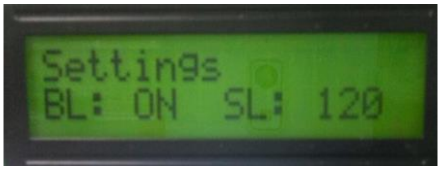
Figure I.3.1

Figure I.3.1 is the display format for internal system time and date.

MON	→ Month
DAY	
YR	→ Year
18:36:06	→ Current Time (24-Hour format)

I.4. Settings

Settings display for Backlight (ON|OFF) and Sleep Timer (in seconds).



Settings
BL: ON SL: 120

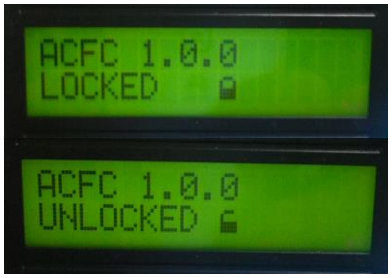
Figure I.4.1

Figure 1.4.1 is the format display for settings.

BL → Backlight (ON|OFF)
SL → Sleep Timer (in seconds)

NOTE: *Feature I.3 and I.4 can be set through the device's controls (Refer to II— Controls).*

I.5. Lock/Unlock



ACFC 1.0.0
LOCKED

ACFC 1.0.0
UNLOCKED

Figure I.5.1

Lock and Unlock feature – discussed on II – Controls *(II.5 – Lock)*

II. Controls

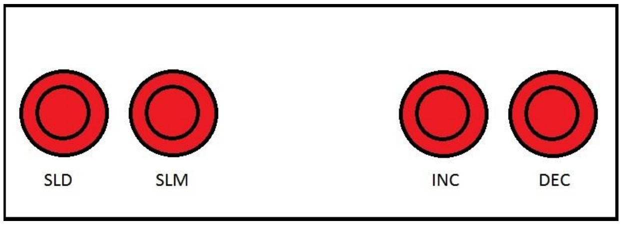


Figure II.1.1

Figure II.1.1 are the controls in front view

II.1. Select Display (SLD)

Selects what data should be displayed. (Refer to I – Features). Data displayed are cleared depending on what sleep time duration is set (Sleep in 120 seconds – default time) for saving battery power. Though the display is in sleep mode, its internal features (e.g. Timer and Counter) are still functioning.

Button Loops
1st Button Switch Count → Fluctuation Timer and Counter
2nd → Total Fluctuation Counts
3rd → System Time and Date
4th → Settings
5th → Go back to 1st Count

II.2. Select Mode (SLM)

This is the button for enabling/disabling for editing the System Time and Date and Settings.

Button Loops
System Time and Date:
1st Button Switch Count → Enables editing Month
2nd → Day
3rd → Year
4th → Hour
5th → Minute
6th → Seconds
7th → Disables editing System Time and Date

Settings:
1st Button Switch Count → Enables editing Backlight
2nd → Sleep Timer
3rd → Disables editing Settings

II.3. Increment (INC)

Control button for editing System Time and Date and Settings. (Increases a unit by 1)

II.4. Decrement (DEC)

Same function as Increment, but decreases a unit by 1.

II.5. Lock

This switch enables the device to be lock/unlock for its controls and displays. This switch is triggered through key combos by pressing SLD and SLM buttons at once (SLD+SLM). Pressing both buttons again will unlock the system.

II.6. Clear

This control puts the device into sleep mode. Like *Lock* switch, it is triggered through key combos by pressing SLD and INC buttons at once. (SLD+INC)

II.7 Reset

This control resets the device's memory. It can be triggered by key combos, SLD and INC.

NOTE: *System Time and Date and Settings can be also set when the device is connected to the computer through Arduino IDE. (Refer to III—Source Code)*

III. Source Code

AC Fluctuation Timer and Counter is run by Gizduino ATmega328P (primarily based on Arduino UNO) that can be programmed through the use of Arduino IDE. This device was programmed on Arduino IDE Version 1.0.6. (Refer to http://www.arduino.cc)

III.1 Source Code Modules

ACFC_MAIN – Contains the main program of the system

DISPLAY_FUNCS – Display functions using LCD 16x2

PROG1 – Contains functions for Fluctuation Timer and Counter

PROG2 – Functions for Total Fluctuation Counts

PROG3 – System Time and Date Functions

PROG4 – Functions for displaying settings

SERIAL_FUNCS – Functions for the device when connected to a PC through the serial monitor installed on Arduino IDE.

SW1 – Select Display Functions

SW2 – Select Mode Functions

SW3 – Increment Functions

SW4 – Decrement Functions

SW_LCK – Lock/Unlock Functions

SW_CLR – Clear Functions

SW_RST – Data reset

SYSTEM_FUNC – Contains the important functions used by the system. (e.g. relay trigger)

SYSTEM_INIT – Functions of the device when initializing

TIMER_FUNC – Primarily composed of timer functions used by the device: Display refresh rate, Sleep Timer & Edit Timer functions.

III.2 Libraries Used

LiquidCrystal.h – Library used for running LCD screen 16x2

avr/pgmspace.h – Used for saving data in the device's flash memory instead of its SRAM

MemoryFree.h – Used for monitoring SRAM remaining

StopWatch.h – Library of functions for accurate Display Refresh timer

inttypes.h – Used for more efficient declaring of variables

EEPROM.h – Library used for saving some data into device's EEPROM (primarily settings data)

Time.h – Set of functions used for computing systems internal date and time

Misc.h – Set of useful functions used by the device (e.g. creating push button switch)

III.3 Setting up Gizduino

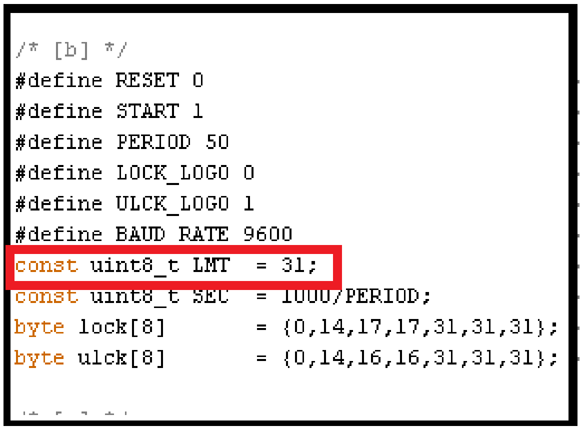
Gizduino ATmega328P uses PL2303 IC (USB to Serial converter) as its interface/medium of programing. In order to successfully program this microcontroller, you need to install PL2303 Prolific Driver (Can be found in AC Timer and Counter/Peripherals Folder).

Current Arduino IDE version used was 1.0.6, as Gizduino doesn't have any patch yet for the current version – 1.6.0

To install Gizduino's patch, just copy the whole "-gizDuino" folder (can be found from AC Timer and Counter/Peripherals/gizDuinoPatch Folder) to Arduino 1.0.6 root folder/hardware. Copy all files from Libraries folder to Arduino IDE root folder/libraries.

III.4 Fluctuation Limits

As default, AC Fluctuation Timer and Counter can take note up to 30 fluctuation events. You can edit its maximum capacity by altering "LMT" variable that can be found on Main System Variables. (Max Fluctuation = LMT – 1)



```
/* [b] */
#define RESET 0
#define START 1
#define PERIOD 50
#define LOCK_LOGO 0
#define ULCK_LOGO 1
#define BAUD_RATE 9600
const uint8_t LMT   = 31;
const uint8_t SEC   = 1000/PERIOD;
byte lock[8]        = {0,14,17,17,31,31,31};
byte ulck[8]        = {0,14,16,16,31,31,31};
```

Figure III.1 – LMT Variable

WARNING: *The device might not work properly if LMT is set beyond SRAM's capacity. To know the current SRAM remaining, connect it to a PC with Arduino IDE then type "-INF" to the Serial Monitor. (Refer to* ***Serial Controls****)*

NOTE: *As specified on its source code, you can only edit Variable Sets A, 1 and 3. Editing other variables might make the system unstable.*

III.5 Serial Controls

Another feature of this device is that, you can communicate to it through computer via Serial monitor found on upper right hand corner of Arduino IDE.

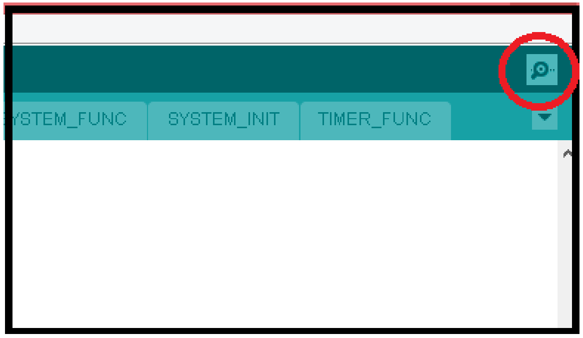
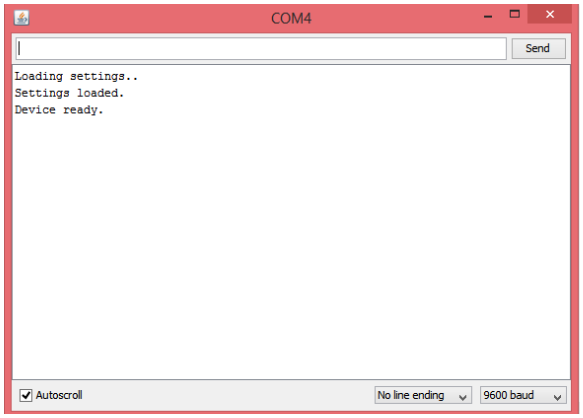


Figure III.2 – Serial Monitor



```
Loading settings..
Settings loaded.
Device ready.
```

Figure III.3 – Serial Monitor Interface

Commands:

-RST : Resets device's EEPROM

-DBG : Puts device into debug mode

-MEM : Displays SRAM remaining

-BL0 : Turns OFF Backlight

-BL1 : Turns ON Backlight

-INF : System Information

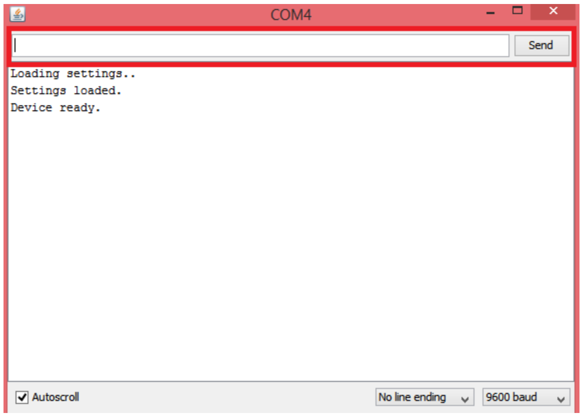
-DEF : Sets settings to default

-SLP SSS : Sets sleep timer (3-digit format; e.g. -SLP 050 → sets to 50 seconds)

-EDT EEE : Sets edit timer (3-digit format; e.g. -EDT 025 → sets to 25 seconds)

The limit for SLP and EDT commands are both 255 seconds

You can type in the commands on the Serial monitor line.



```
Loading settings..
Settings loaded.
Device ready.
```

Figure III.4 – Serial Monitor Command Line

NOTE: *To use the following commands: -RST, -BL0, -BL1, -DEF, -SLP SSS and -EDT EEE, the user must set the device first into debug mode, -DBG.*

III.6 Uploading the codes

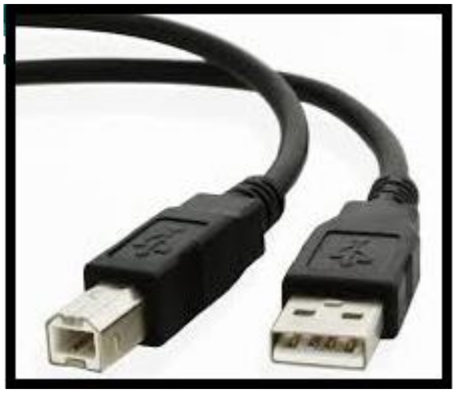
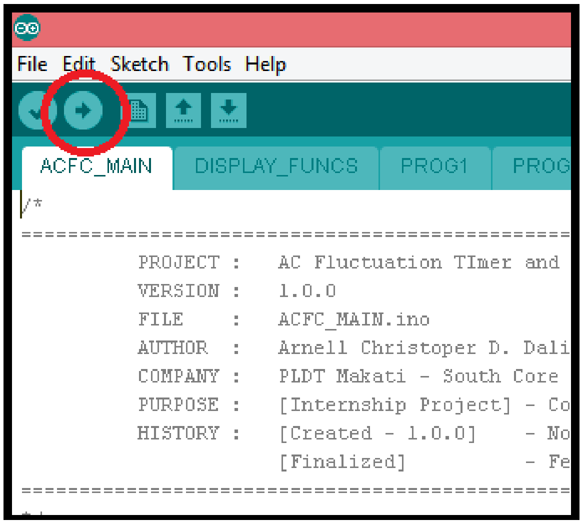


Figure III.5 – Printer USB connector

Gizduino uses Printer USB connector for programming.

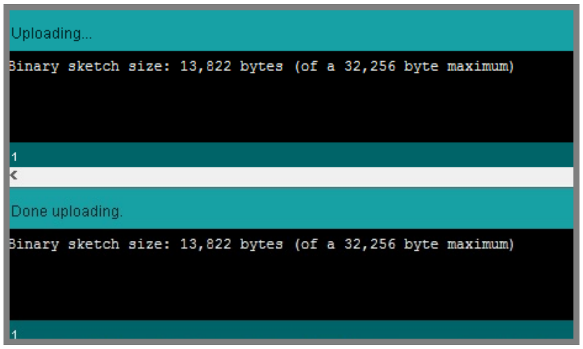
NOTE: *Refer to* ***Gizduino.pdf Manual – Figure 3*** *on how to connect Gizduino to PC.*



```
/*
=============================================
          PROJECT :   AC Fluctuation TImer and
          VERSION :   1.0.0
          FILE    :   ACFC_MAIN.ino
          AUTHOR  :   Arnell Christoper D. Dali
          COMPANY :   PLDT Makati - South Core
          PURPOSE :   [Internship Project] - Co
          HISTORY :   [Created - 1.0.0]     - No
                      [Finalized]           - Fe
=============================================
```

Figure III.6 – Upload Code

To upload the code, click the arrow button that can be found on the left-hand corner of the IDE.



```
Uploading...
Binary sketch size: 13,822 bytes (of a 32,256 byte maximum)
1

Done uploading.
Binary sketch size: 13,822 bytes (of a 32,256 byte maximum)
1
```

Figure III.7 – Done Uploading

When the user successfully uploaded the code, a notice will be denoted by Arduino IDE – “Done uploading.” that can be found at the lower left part of it.

IV. Block Diagram

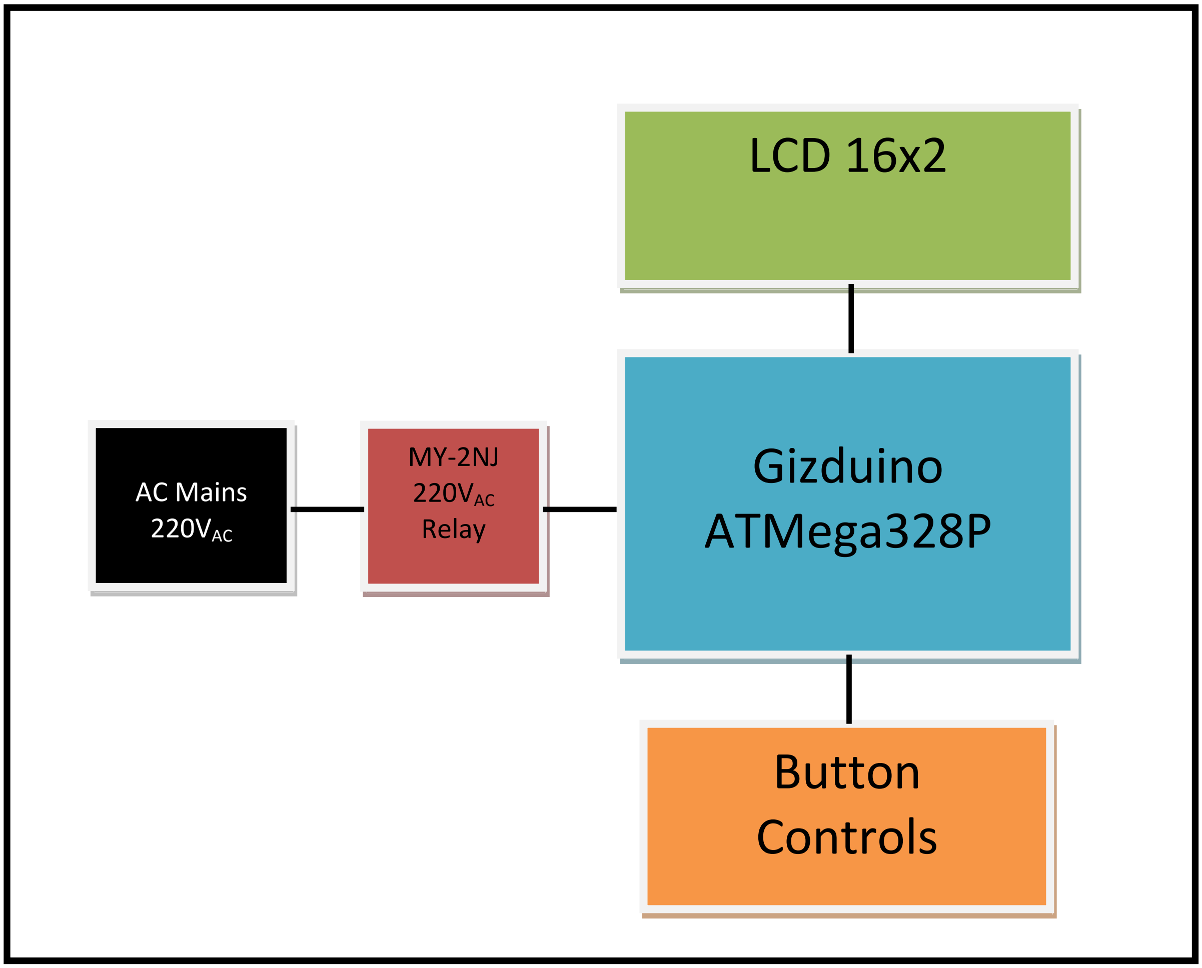


Figure IV.1 – Block Diagram

The above figure shows the block diagram of the device. Gizduino ATmega328P acts as the main CPU and controller of the device. It processes all data in real-time, which include current relay state and the system's time and date. LCD 16x2 and the Button Controls act as the user-interface of the device. Through the Button Controls, the user can choose what to be displayed in the LCD; edit the time and date and/or change the settings as preferred by the user. As discussed on Chapter 1, the LCD can display four sets of data. MY-2NJ 220V$_{AC}$ Relay functions as the main sensor of the device and AC Mains acts as its stimulus. Since the relay is configured as NC (Normally-Closed), as the commercial power goes down (AC Mains), the relay is de-energized, thus sending a signal to Gizduino. It will then process the signal and start a stopwatch timer to compute for the duration of commercial power downtime. If AC Mains goes back, the microcontroller will stop the timer and save the data gathered (*Refer to I.1 – Fluctuation Timer and Counter*).

V. Schematic Diagram

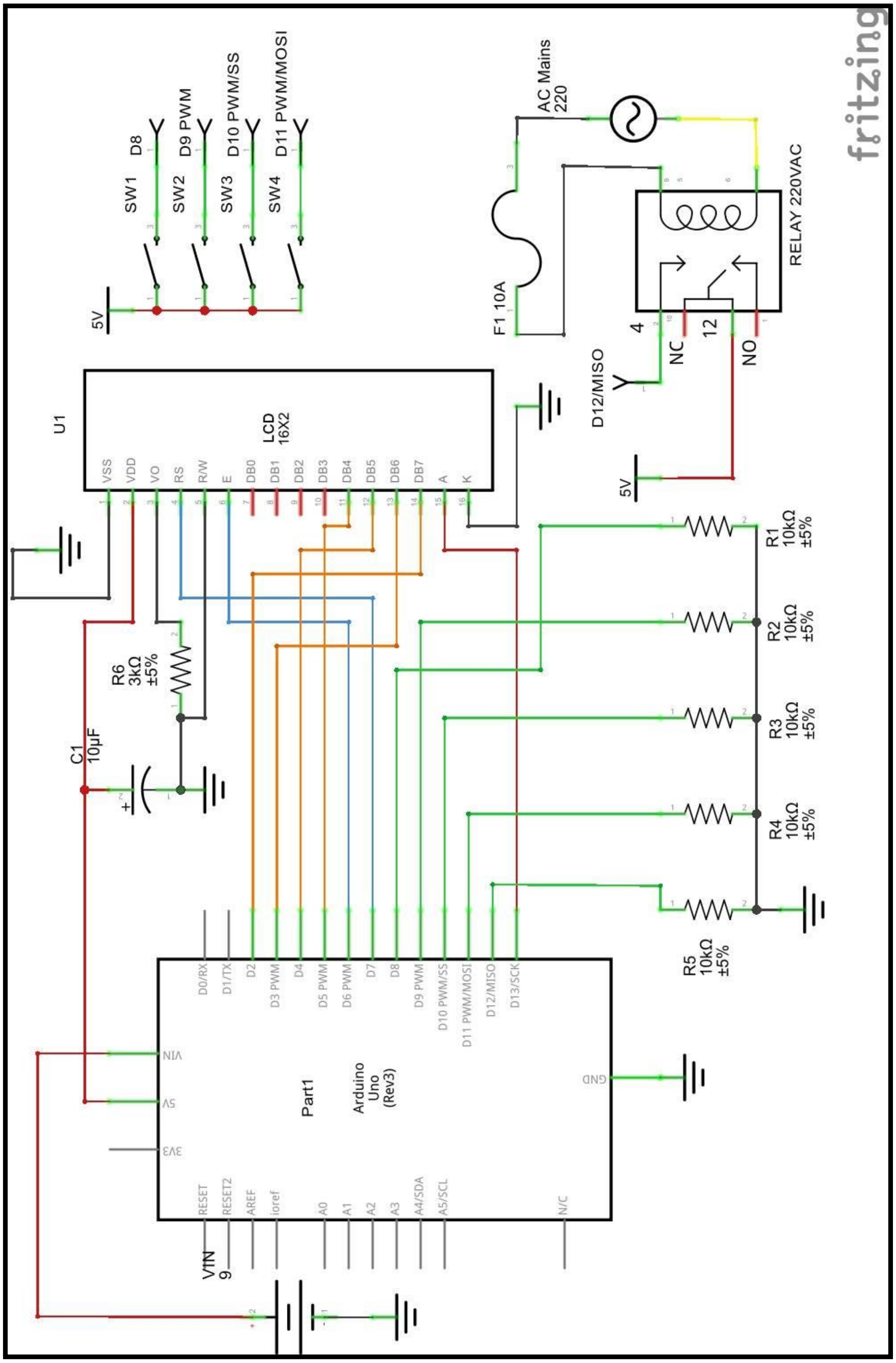


Figure V.1 – Schematic Diagram

NOTE: *Arduino Uno (Rev3) → Gizduino ATmega328P*

VI. Layout

VI.1 Button Mount PCB

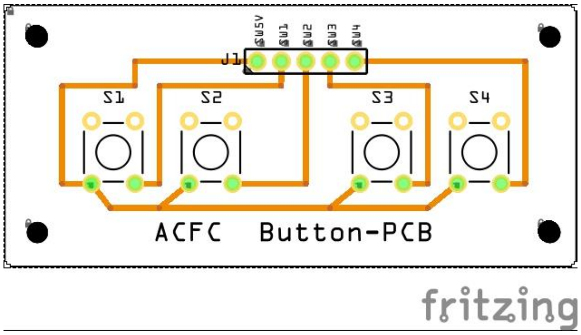


Figure VI. – Button Mount (Top View)

VI.2 Gizduino Mount PCB

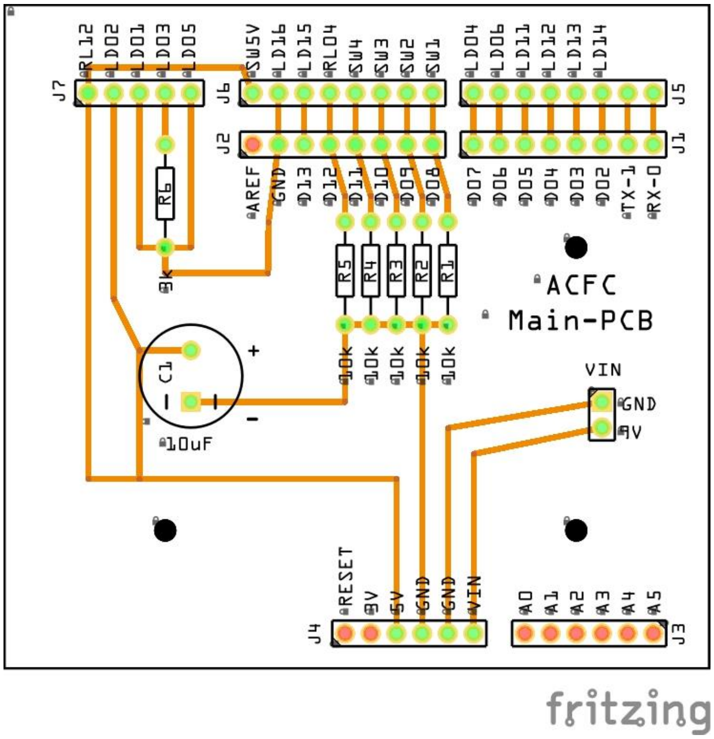


Figure VI. – Gizduino Mount (Top View)

LEGEND: **LD -> LCD**
RL -> Relay
SW-> Switch
R -> Resistor
C -> Capacitor

VII. Parts List

COMPONENT	PRICE	QUANTITY	TOTAL PC/COMPONENT
Gizduino ATmega328P	662.00	1	662.00
LCD 16x2 (Hitachi HD44780 Driver Compatible)	200.00	1	200.00
OMRON MY-2NJ 220VAC Relay	150.00	1	150.00
1-PIN Female-to-Female Connector	5.00	12	60.00
4-PIN Female-to-Female Connector	20.00	2	40.00
9V Battery Snap Connector	5.00	1	5.00
9V Battery	150.00	1	150.00
8-PIN Header (Female)	2.50	4	10.00
6-PIN Header (Female)	2.00	2	4.00
40-PIN Header (Male)	12.50	1	12.50
5-PIN Header (Male-Long)	0.25	1	0.25
Tact Switch	2.50	4	10.00
Resistor 3kΩ	0.25	4	1.00
Resistor 10kΩ	0.25	5	1.25
Solid Wire (Quantity in meters)	5.50	3	16.50
PCB 60x28mm	5.00	1	5.00
PCB 66x70mm	15.00	1	15.00
AC Power Plug	40.00	1	40.00
Panel Mount Fuse	10.00	1	10.00
Fuse 10A	20.00	1	20.00
Capacitor 10uF	5.00	1	5.00
	TOTAL		1417.50

Table VI.1 – Parts and Price List

OTHERS			
Item	Price	Quantity	Subtotal
Glossy Paper (per 20 pcs)	50.00	1	50.00
Ferric Chloride (4 oz)	100.00	1	100.00
		TOTAL	150.00

Table VI.2 – Miscellaneous

BIBLIOGRAPHY

ATmega48A/PA/88A/PA/168A/PA/328/P. (2014, October). Retrieved November 20, 2014, from Atmel: http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf

Building Circuit. (n.d.). Retrieved February 24, 2015, from Fritzing: http://fritzing.org/building-circuit/

Enrison, R. (2013, March 22). *Episode 2 Arduino and Pinguino Clones.* Retrieved November 20, 2014, from e-Gizmo Technical Blog: http://e-gizmo.blogspot.com/2013/03/whats-cooking-e-gizmo-episode-2-arduino.html

Enrison, R. (2013, April 24). *Patch Files to add gizDuino.* Retrieved November 20, 2014, from e-Gizmo Technical Blog: http://e-gizmo.blogspot.com/2013/03/patch-files-to-add-gizduino-in-arduino.html

General Purpose Relay MY. (n.d.). Retrieved November 20, 2014, from Industrial Omron: http://industrial.omron.com.br/uploads/arquivos/MY.pdf

Gizduino Compatible Kit. (n.d.). Retrieved November 20, 2014, from Google Docs: https://docs.google.com/file/d/0BxdLxDCD6Hidem1aQVlCcEhocDQ/edit?pli=1

Gizduino Installation Guide. (n.d.). Retrieved November 20, 2014, from Google Docs: https://docs.google.com/file/d/0BxdLxDCD6HidMlQ2bmk4U00xQ3c/edit?pli=1

Liquid Crystal. (n.d.). Retrieved November 20, 2014, from Arduino: http://arduino.cc/en/Reference/LiquidCrystal

Margolis, M. (2009). *Time.* Retrieved November 20, 2014, from Arduino: http://playground.arduino.cc/Code/Time

Murdoch, M. (n.d.). *Memory Free.* Retrieved November 20, 2014, from Arduino: http://playground.arduino.cc/code/AvailableMemory

Tillaart, R. (n.d.). *Stopwatch.* Retrieved November 20, 2014, from Arduino: http://playground.arduino.cc/Code/Stopwatch

Toner Management. (n.d.). Retrieved March 5, 2015, from Visionics: http://www.visionics.a.se/html/pdf/tut/Tonner%20transfer%20method%20for%20PCB%20manufacturing.pdf

Windows Driver Download. (n.d.). Retrieved November 20, 2014, from Prolific: http://www.prolific.com.tw/US/ShowProduct.aspx?p_id=225&pcid=41