Talking Alarm Clock

User/Set-Up Manual

Powered by: WIZ750SR-232





Table of Contents

| _ | TOPIC | pages |
|-------|--|----------|
| | | |
| Proje | ect Description | |
| • | General Description of Project. | 3-4 |
| • | | |
| Set-L | Jp Components | |
| • | WIZ750SR-232 Module Software / Hardware. | 5-7 |
| • | Gen4-uLCD-32PT Display Program Firmware. | 8-11 |
| • | MP3 Player Shield – Install MP3 files. | 12 |
| Syste | em Hardware Information | |
| ٠ | Parts List. | 13 |
| • | Electrical Hardware Print. | 13 |
| Set-l | Jp Arduino MEGA 2560 R3 board. | |
| • | Upload Firmware. | 14 |
| Ardu | ino Software Routines. | |
| • | WIZ750SR-232. | 15-17 |
| • | Gen4-uLCD-32PT. | 18 |
| ٠ | MP3 Plaver Shield. | 19-20 |
| • | RTC DeadOn. | 21-23 |
| • | Multi-Purpose Pushbutton. | 24-25 |
| ∆larr | n Clock Interface Software on PC | |
| • | Software Installation | 26-27 |
| • | Default Setup File (setupdata ini) | 20 21 |
| • | PC ann overview | 28 |
| • | Set Current Date/Time | 20 |
| • | Appounce Time at the Top of the Hour | 20 |
| • | Soloct Clock Background Scroops | 31 |
| • | Select Clock Dackground Screens. | 20 |
| • | Set Lin Alarm Time | J∠ 22 |
| • | Set-Up Alarm and Sereene | 33 24 |
| • | Set wake-up Alarm and Screens. | 34 |
| • | Play Alalin Tracks. | 35 |

TAC = Talking Alarm Clock

Project Description: General Description of Project.

This project, the Talking Alarm Clock (TAC) keeps accurate time, features a large display for viewing time and incorporates numerous alarm wake-up functions. It's built around the WIZ750SR module which allows the user to customize the alarm clock to suit their needs/preferences.

The time is announced with recordings of my daughter when she was 4 years old back in 2000. I waited until now to make a project that allowed MP3 audio files to be played easily and to connect the clock to the LAN!

Customizable Features:

- 1. Wake-up alarms consists of 16 different MP3 audio files that can be customized.
 - Add your own MP3 audio tracks to sound wake-up alarms.
 - Add your own MP3 audio tracks to announce times.
- 2. The current time can be announced anytime by pressing a button or at the top of the hour.
- 3. PC app to communicate with the TAC using the WIZ750SR module making it an ease to:
 - Set the current date and time with just a click of a button.
 - Test and play the alarm audio files for just the right wake-up alarm.
 - Set the desired wake-up time along with alarm audio.
 - Three selectable wake-up screens.
- 4. All entered values are stored in the RTC battery backed up ram so you will never lose your settings even after a power failure.

The components used in this project consists of: WIZ750SR Serial to Ethernet module, Arduino Mega 2560 R3 board, Real-Time clock board, MP3 shield board and a Gen4-uLCD-32PT LCD Display. Set-up of each board will be discussed in detail in chapters to follow.

A 12vdc power supply provides power to the Arduino Mega board which converts this into 5vdc and 3.3vdc power. The 3.3vdc is used to power the WIZ750SR module and RTC board. The 5vdc provides power for the MP3 board and display. The Arduino Mega communicates with the WIZ750SR module and the LCD display using serial communication (a separate port for each device) and the MP3 and RTC boards board using the SPI communication protocol. The PC app communicates with the TAC using the WIZ750SR Serial to Ethernet module as a gateway to the local area network or over the internet. An ethernet cable is run to the TAC for this purpose. With this connection, the TAC can be accessed anywhere a PC can connect to the internet.

The block diagram below illustrates the connections between the project's main components.

Block Diagram:



Photos 1 and 2 depict the completed assembly. All components are soldered together and use jumper wires. The boards are is fastened inside the enclosure and the only connections required to make the boards functional is a 12vdc power supply. Attach an ethernet cable to program features.



Photo 2: Completed assembly showing component layout and wiring connections.



WIZ750SR-232 (Ethernet to Serial Communication): Software Set-Up.

Power up the WIZ750SR-232 module and connect to your local area network.



Download the WIZnet S2E Configuration Tool Program: https://github.com/Wiznet/WIZnet-S2E-Tool-GUI/releases

References:

WIZ750SR Getting Started Guide: https://wizwiki.net/wiki/doku.php?id=products:wiz750sr:gettingstarted:start

WIZnet S2E Configuration Tool Program: https://github.com/Wiznet/WIZnet-S2E-Tool-GUI/releases

WIZ750SR Product Overview: https://wizwiki.net/wiki/doku.php?id=products:wiz750sr:overview:en

WIZ750SR Datasheet: https://wizwiki.net/wiki/doku.php?id=products:wiz750sr:datasheet:start

Run the WIZnet S2E Configuration Tool Program:

| Wiznet 52t Configuration Tool 0.5.3 de | V | | |
|--|--------------------------------|---|------------------------------------|
| | | | |
| Search Setting | Upload 🔯 Reset 🕻 | Factory Factory Config | Load Config |
| Mac addr Name | Settings Options | 1 | I |
| 1 00:08:DC:54:88:25 WIZ750SR | Device info | Status pin | Serial debug |
| | Firmware version 1.2.0 | PHY link TCP conr DSP | Enable - |
| | Network configuration | | |
| | 102 168 0 24 | IP allocation | TCP timeout |
| | 192.108.0.24 | 4 Static DHCP | Retry count 8 |
| | Subnet mask 255.255.255.0 | Search identification code | |
| | Gateway 192.168.0.1 | Code | Max. 8 bytes) Show chars |
| | DNS server 209 18 47 62 | Connection password (TCP se | erver mode only) |
| | | Enable Password | Show chars |
| | | | |
| | Channel #1 Channel #2 Chann | el #3 Channel #4 | |
| | Status: OPEN | - Serial options | Serial command mode |
| | Serial interface: RS-232/TTL | Baud rate 115200 4 🔻 | Triger code 2B 2B 2B |
| | Operation mode | Data bit 8 | Timer intenal |
| | TCP client | | |
| Search ID code | TCP server | | The connection holding period when |
| ID code | TCP mIxed (server/client) | Stop bit | data transmission. |
| Show chars | © UDP | Flow control NONE | Reconnection Interval 3000 ms |
| Search method | Local port | | |
| OUDP broadcast | Local Port 5000 | Serial data packing condition | TCP Keep-Alive |
| TCP unicast | Remote IP / port | Character 00 | 🗹 Enable |
| 192.168.11.2 : 50001 | Remote ID 102 169 11 2 | Size 0 | Initial Interval 7000 ms |
| MAC address | Remote 19 192.100.11.3 | Time(ms) 0 | Retry Interval 5000 ms |
| | LIGBOOTO DOFT LIGHT | I III I | |

- 1. Click on the "Search" button.
- The Mac address of your module should appear in the display list below the "Search" button. Click on your "Mac address" and the module configuration will appear to the right. Note: Mac address can be found on sticker attached to module.
- Record the local IP address. For this example is address is 192.168.0.24.
 "DHCP" will be selected 1st time you run the program. See page 32 to use IP address in PC app.
- 4. Click on:
 - "Static" option button.
 - "Baud rate drop down box and select "9600".
 - "Parity drop down box and select "NONE".
- 5. Click the "**Setting**" button to save new module configuration.
- 6. Click on the "**Search**" button again to display the new module configuration to the right.
- 7. Module configuration is now complete, close program.

WIZ750SR-232 (Ethernet to Serial Communication): Hardware Set-Up.

From the Print on page 13.

Arduino Mega connections: (serial port #1) Pin 18 to MAX233 pin 2 Pin 19 to MAX233 pin 3 Pin 46 to WIZ750SR nRESET

MAX233 connections: Pin 5 to WIZ750SR pin RXD Pin 4 to WIZ750SR pin TXD

Power pins: 3.3vdc to WIZ750SR pin VCC Ground to WIZ750SR pin GND



Set-Up 4D System Gen4-uLCD-32PT LCD Display. Program/Upload Firmware.

SK-gen4-32PT (Starter Kit) includes the following:

- gen4-uLCD-32PT (Resistive Touch) Display Module
- gen4-IB (Interface Board)
- 150mm FFC Cable
- 4D-UPA (Programming Adaptor)
- 5-way F-F Ribbon Cable w/ M-M Adaptor
- 4GB microSD Card
- Quick Start Guide

Gen4-uLCD-32PT Intelligent Display Module: https://www.4dsystems.com.au/product/gen4_uLCD_32PT/

<u>4dsystems/ViSi-Genie-Arduino-Library:</u> https://github.com/4dsystems/ViSi-Genie-Arduino-Library



- 1. Download the 4D Systems' Workshop4 IDE as per instruction in Workshop4 IDE INSTALLATION GUIDE.
 - Download and Install the 4D Systems' Workshop4 IDE: <u>https://www.4dsystems.com.au/product/4D_Workshop_4_IDE</u>
- 2. Power Up and Connect to a Computer Using 4D-UPA:
 - Connect one end of the FFC to your module's 30-way ZIF socket with the metal contacts on the FFC facing up on the latch.
 - Connect the other end of the FFC to the 30-way ZIF socket on the 4D-UPA with the metal contacts on the FFC facing up on the latch.
 - Connect the USB-Micro-B Cable to the 4D-UPA.
 - Lastly, connect the other end of the USB-Micro-B Cable to the computer.
 - The display will indicate that the disk in not mounted.



 Insert the micro-SD card into a micro-SD to SD card converter and plug the SD card converter into the SD card slot of the PC



- The software will download the program to flash on the LCD and SD card.
- 3. Start Using Workshop4 IDE
 - Open Workshop4 IDE and Open an existing project.



- Click on "**Comms**" tab.
- Com port will be displayed and the indicator will turn blue.

| Workshop 4 - aclock(Gen4-uLCD-32PT, LANDSCAPE_R) | | | | | | |
|--|--------------------------------------|--|--|--|--|--|
| File Home View Tools Comms Project | K AND AND AND AND AND AND AND AND A | | | | | |
| COM 8 Status at 5:51:08 PM The system cannot find the file specified opening Com Port Comms | 4 Þ | | | | | |
| | 2 Aking Tanangan | | | | | |
| | | | | | | |
| Talking Alarm Clock by ABEtronics 4-25-18 ver1.0 | Form Form0 | | | | | |
| | Object Form0 | | | | | |
| | Properties Events | | | | | |
| | Property Value Name Form0 | | | | | |
| | Alias Form0 | | | | | |
| | Bgtype Color | | | | | |
| | Color BLACK | | | | | |
| | T Source | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| Insert | Press F1 for context sensitive help: | | | | | |

- Click on "Project" tab.
- Click on "Run Flash" button, Select 115200 Comms speed.
- Select Display: Landscape-R

| File Home View Tools Comms Project | |
|--|--|
| Comms Speed: 115200 Allow buttons Run Form0 Sound Buffer: 1024 Allow buttons | Allow -ve Led and Custom Digits and leading blans on Custom Digits |
| Genie | Enhancements Display |
| Sadock X | 4 Þ |
| Form0 Øbject Inspe | sctor 🕄 |
| Form Form | n0 🗸 |
| Object For | m0 🔹 |
| Properties | Events |
| Property | FormD |
| Alias | Form0 |
| Bgtype | Color |
| Color | BLACK |
| Image I Source | (None) |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| Insert | Press F1 for context sensitive help |

| Click on "Home" tab. | | | | _ |
|--|----------------------------|-----------------|-------------------|-------------------------|
| Click on "Build" button | | | | |
| Workshop 4 - aclock(Gen4-uLCD-32PT, L Rie Home View Tools Comms Project | ANDSCAPE_R) | Build | (Build) | |
| New Open Save Ac Print Savid CopyLload Build CopyLload Build CopyLload Build CopyLload Build CopyLload Copylload (CopyLload CopyLload Co | Gauges I/O | Copy/Load Bu | Copy/Load iild | |
| Farmit 8 | Object Inspecto | ar | | 8 |
| Talking Alarm Glock by ABEtronics 4-25-18 ver1.0 | Form Form0 Object Form0 | | | |
| | Property | Value | | |
| | Name Alias | Form0 Form0 | | |
| | Bgtype | Color | | |
| | Color | (None) | | |
| DM | E Source | (increase) | | |
| | | | | |
| | | | | |
| insert | | | Press F1 for co | ontext sensitive help 🖽 |

• The Display and SD card will now be programmed.

4. Insert the SD card in the display module.

• Remove the uSD Card from your computer and insert it into the uSD Card slot on the gen4 HMI Display module.

*NOTE: When the uSD card is not yet inserted, this message will appear on the gen4 HMI Display Module: "Drive not mounted". After inserting the uSD card the GUI should appear on the gen4 Display.



Set-Up Components: MP3 Player Shield – Install MP3 files.

MP3 Player Shield Hookup Guide V15: MP3 Player Shield Hookup Guide

MP3 File and µSD Card Setup:

 The SDFat Arduino library we'll be using only supports cards formatted for FAT16 or FAT32. Your card is probably already formatted to one of these standards, but it doesn't hurt to double check.

Prepare your Music Files:

- The VS1053B is capable of playing MP3, AAC, WMA, MIDI, and Ogg Vorbis audio files. The VS1053B supports a variety of sample rates and bitrates for each file type. MP3's, for example, are supported at up to a **320 kbps bitrate and a 48 kHz sample rate.**
- The SDFat library only supports "8.3" file names that's eight characters before the '.' and three characters after (e.g. "track001.mp3").
- Create MP3 audio files. (MP3 file and uSD card set-up)
 - Track Names:
 - Track001 059 = 1 to 59. Track101 – 109 = 01 to 09. Track150 – 165 = alarm audio clip #1 to #16. Track200 = The Time Is. Track201 = AM. Track202 = PM. Track210 = Have a Nice Day. Track211 = O-Clock. Track220 = Good Morning. Track221 = Good Morning. Track222 = Good Evening. Track223 = Good Evening. Track224 = System Ready. Track225 = Hello.

Using the SFEMP3Shield Library:

- SFEMP3Shield greatly simplifies the task of interfacing with the VS1053 and using the MP3 Player Shield.
- The "Sparkfun-MP3-Player-Shield-Arduino-Library-master" folder you extract should have a handful of folders within. The "SFEMP3Shield" and "SdFat" folders in particular need to be installed as Arduino libraries.

System Hardware Information: Parts List.

| Qty | PART # | Description |
|-----|----------------------------------|---|
| 1 | WIZnet: WIZ750SR-232 | WIZnet Serial to Ethernet board |
| 1 | SparkFun: Arduino Mega 2560 R3 | Microcontroller board |
| 1 | SparkFun: MP3 Shield board | MP3 player |
| 1 | DS3234 | SparkFun DeadOn RTC Breakout |
| 1 | 4D Systems: Gen4-uLCD-32pt | LCD display |
| 1 | SparkFun: PRT-00337 | Coin Cell Battery - 12mm (CR1225) |
| 1 | Mouser: MAX233CPP | Transceiver chip (RS-232 Interface) |
| 1 | SparkFun: COM-10439 | LED Tactile Button- White |
| 1 | SparkFun: BOB-10467 | SparkFun LED Tactile Button Breakout |
| 1 | SparkFun: COM-13833 | SparkFun microSD Card with Adapter - 16GB (Class 10). |
| | | For MP3 player. |
| 1 | 10k ohm resistor | |
| 1 | 100 ohm resistor | |
| 1 | 12vdc power supply | |
| 2 | SparkFun: Speakers 0.25W, 8 ohms | COM-09151 |
| 1 | 6ft ethernet cable | |
| 1 | Mouser: 20 pin dip socket | |
| 1 | Mouser: 563-CU-3283 | Mouser Enclosure |
| 1 | RTL-11302 | USB-A to B Cable |
| 1 | SparkFun PRT-14284 | Jumper Wires Premium 4" M/M - 26 AWG (30 Pack) |
| 1 | SparkFun PRT-12794 | Jumper Wires - Connected 6" (M/F, 20 pack) |
| 4 | 1/8" LCD mounting screws | |

Electrical Hardware Print.



Set-Up Arduino MEGA 2560 board: Upload Firmware.

- 1. Connect Arduino board to PC using USB cable.
- 2. Make new sketch folder under Arduino in My Documents.



- 3. Open the "aClock.ino" file.
- 4. Click on the "upload" button.

| 🥯 aClock Arduino 1.0.5-r2 | |
|--|---|
| File Edit Sketch Tools Help | |
| 👽 📀 🗈 🔛 Upload | P |
| aClock | |
| // // Project: Jackie Talking # // Date: 10-1-14 // Author: Daddy // | Alarm Clock by ABEtronics 2015. |
| // // Set-Up Libraries // | |
| #include <\$PI.h> | // SPI library |
| <pre>#include <sdfat.h></sdfat.h></pre> | // SDFat Library |
| #include <sdfatutil.h></sdfatutil.h> | // SDFat Util Library |
| #include <softwareserial h=""></softwareserial> | // mps snield Library |
| SdFat sd; | // Create object to handle SD functions |
| SFEMP3Shield MP3player; | // Create Mp3 library object |
| SoftwareSerial LCD_Serial(0, | .5); // LCD Serial Communication TX onl |
| < | |
| | |
| | |
| 1 | Arduino Uno on COM8 |

Arduino Software Routines: WIZ750SR-232.

Set-up serial port #1 parameters:

Check Serial port #1 for activity:

```
_____
// check Serial communication: if character received
void Check Serial Com()
{
 if (Serial1.available() > 0)
 {
   getsercom(Serial1.read());
                              // get command from serial input
 }
}
// _____
// Check Serial Port for Incoming Characters
// -----
void getsercom(byte key command)
{
 // MP3 file descriptions (track###.mp3 where ## = 000-255)
 // 1-59 1-59
// 01-09 101
 // Alarm Tracks = 109
// Alarm Tracks = 150 - 170
// The Time Is = 200
// AM =
 // AM =
 // PM = 202
// have a nice day = 210
 // o-clock = 211
// no = 219
 // good morning = 220
// good afternoon = 221
 // good evening = 222
 // good night = 223
 // Sys Ready =
// Hello =
                     224
 // Hello =
                    225
 //-----
 // Commands:
 // T = Announce Current Time format: T, hr(1-12), min(1-59), ampm(1-2)
 11
                                               min(0) say o-clock
 // H = Announce Current Time Hourly
 // h = Don't Announce Current Time Hourly
 // A = Enter Wake Up Alarm Time format: A, hr(1-12), min(1-59), ampm(1-2), track#, vol
  // P = Play Alarm MP3 Tracks format: P,track#,vol

      // S = Stop Track
      format: S

      // E = Enable alarm mode.
      format: E

      // D = Disable alarm mode.
      format: D

      // ! = Test Alarm.
      format: T (Play Alarm and disable again)

 // 1 = background #1
 // 2 = background #2
 // 3 = background #3
 // 4 = background #4
 // 5 = wake up screen #1
 // 6 = wake up screen #2
 //7 = wake up screen #3
```

Example of each serial command:

```
//--
// Backgrounds
//-----
if (key command == '1')
{
 genie.WriteObject(GENIE_OBJ_FORM, 0, 1); // change startup screen (#,1 #=form#)
 background = 1;
 WriteBackgroundtoRTC (background);
 Read Display Time();
 return;
//-----
                 _____
// Wake Up Screens
_____
if (key_command == '5')
{
 genie.WriteObject(GENIE OBJ FORM, 5, 1); // change startup screen (#,1 #=form#)
 wakeupscreen = 1;
 WriteWakeupScreentoRTC (wakeupscreen);
 return;
//-----
// Enable/Disable Wake Up Alarm
_____
if (key command == 'E') {Enable Wakeup Alarm();}
if (key_command == 'D') {Disable_Wakeup_Alarm();}
//-----
// Announce Hourly
//-----
if (key_command == 'H')
{
 announce hourly = 1;
 WriteAnnounceHourlytoRTC (announce hourly);
if (key command == 'h')
{
 announce hourly = 0;
 WriteAnnounceHourlytoRTC (announce hourly);
//-----
// TEST Wake Up Alarm
//-----
                  _____
if (key command == '!')
{
 Enable Wakeup Alarm();
 TEST Wakeup_Alarm = 1;
// -----
                _____
// Set Current Time
// T, month, day, year, hr, min, sec, AmPm
// ----__/
if (key command == 'T')
{
 tti = 0;
 hris = 0;
 minis = 0;
 ampmis = 0;
 hand = 0;
 pmp3f = 0;
 pmp3f = 0;
S_month = Serial1.parseInt(); // 1-12
S_day = Serial1.parseInt(); // 1-31
S_year = Serial1.parseInt(); // 20xx
HR = Serial1.parseInt(); // 1-12
MIN = Serial1.parseInt(); // 1-59, 60=OCLOCK
SEC = Serial1.parseInt(); // 1-59, 60=OCLOCK
AMPM = Serial1.parseInt(); // 1=AM, 2=PM
wel alarm = Serial1 parseInt(); // 1=AM, 2=PM
 vol_alarm = Serial1.parseInt(); // Alarm volume
 vol alarm = 0;
 if (MIN >= 100)
  MIN = MIN - 100;
   // ------
```

```
//day(1-31), month(1-12), year(0-99), hour(0-23), minute(0-59), second(0-59)
   // -----
   SetTimeDate(S day, S_month, S_year, HR, MIN, SEC);
   MIN = MIN + 100;
 } else
   // -----
   //day(1-31), month(1-12), year(0-99), hour(0-23), minute(0-59), second(0-59)
   // ------
 SetTimeDate(S day, S month, S year, HR, MIN, SEC);
 Read Display Time(); //Read Date and time and display on LCD.
 UPDATE LCD();
 pmp3f = 1;
                             //Play mp3 files flag
 good day = 1;
 MP3player.setVolume(0, 0);
 return;
.
// -----
// Set Wake Up Alarm Time
// A = Set Wake Up Alarm Time format: A,hr(1-12),min(1-59),ampm(1-2)
// -----
if (key_command == 'A')
{
 tti = 0;
 hris = 0;
 minis = 0;
 ampmis = 0;
 hand = 0;
 pmp3f = 0;
 alarm flag = 0;
 WAKE_UP_JACK = 0;
 HR = Serial1.parseInt(); // 1-12
MIN = Serial1.parseInt(); // 1-59, 60=OCLOCK
AMPM = Serial1.parseInt(); // 1=AM, 2=PM
 AMPM = Serial1.parseInt(); // 1=39, 00=0CLC
alarm_track = Serial1.parseInt(); // 1=AM, 2=PM
vol_alarm = Serial1.parseInt(); // Alarm Track#
vol_alarm = 0. // Plarm_track#
 vol alarm = 0; // Alarm volume
 if (MIN==0 || MIN==59) {announce hourly=0;} // Turn off annouce hourly is alarm min=0
 // -----
 //Set Wake Up Alarm Time.
 // -----
 HR ALARM = HR;
 MIN ALARM = MIN;
 AMPM ALARM = AMPM;
 RTC_HR = HR;
 RTC MIN = MIN;
 RTC AMPM = AMPM;
 // ------
 //day(1-31), month(1-12), year(0-99), hour(0-23), minute(0-59), second(0-59)
 // ------
 SetAlarmTime(HR, MIN, AMPM, alarm_track, 0);
 WriteAlarmDatatoRTC(HR, MIN, AMPM, alarm track, 0);
 UPDATE LCD();
                           //play mp3 files flag
 pmp3f = 2;
 MP3player.setVolume(0, 0);
 playmp3();
 return;
// ------
// Sample Alarm MP3 Track
// -----
if (key_command == 'P')
{
 song = Serial1.parseInt(); //track #
vol_alarm = Serial1.parseInt(); //Alarm volume
 vol alarm = 0;
 if (song > 100)
 {
  pmp3f = 3;
                             //play mp3 files flag
   MP3player.setVolume(0, 0);
   return;
}}//end routine
```

Arduino Software Routines: Gen4-uLCD-32PT.

Include genie Arduino library:

-----#include <genieArduino.h> // LCD Library Genie:

Set-up serial port #2 parameters:

_____ // #2 serial communication for LCD DISPLAY 115200,8N1 // ------Serial2.begin(115200); // GEN4-uLCD-32PT. genie.Begin(Serial2); // Use Serial1 for talking to the Genie Library, and to the 4D Systems display genie.AttachEventHandler(myGenieEventHandler); // Attach the user function Event Handler for processing events

```
Reset the display at start-up:
 // Reset the Display.
 // -----
 digitalWrite(LCD_Reset, 0); // Reset the Display via D10
digitalWrite(WIZ750SR_nReset, 0); // Reset WIZ750SR (active low)
 delay(100);
 delay (3500);
                            //let the display start up after the reset
```

Change form and background/text colors: (change form & background 1-4)

{genie.WriteObject(GENIE OBJ FORM, 0, 1);}

Change wake-up screens: (change wake-up screens 1-3)

```
genie.WriteObject(GENIE OBJ FORM, 5, 1);}
```

Update current time and alarm time:

// _____ genie.WriteObject(GENIE OBJ LED DIGITS, 0x01, RTC MIN); genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x00, RTC_HR); genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x02, HR_ALARM); genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x03, MIN_ALARM); //RTC 1=AM, 2=PM if (ap == 1) {genie.WriteObject(GENIE_OBJ_STRINGS, 0x00, 1);} // LCD PM=0, AM=1
if (ap == 2) {genie.WriteObject(GENIE_OBJ_STRINGS, 0x00, 0);} // LCD PM=0, AM=1 if (AMPM_ALARM == 1) {genie.WriteObject(GENIE_OBJ_STRINGS, 0x04, 1);} // LCD PM=0, AM=1 if (AMPM ALARM == 2) {genie.WriteObject(GENIE OBJ STRINGS, 0x04, 0);} // LCD PM=0, AM=1

Arduino Software Routines: MP3 Player Shield.

Include libraries:

| // | |
|--|--|
| <pre>#include <spi.h> #include <sdfat.h> #include <sdfatutil.h> #include <sfemp3shield.h></sfemp3shield.h></sdfatutil.h></sdfat.h></spi.h></pre> | // SPI library // SDFat Library // SDFat Util Library // Mp3 Shield Library |
| SdFat sd; SFEMP3Shield MP3player; | // Create object to handle SD functions // Create Mp3 library object |

Initialize the MP3 Player Shield. (Uses SPI Protocol interface)

Play MP3 Files:

```
_____
// Play MP3 Files
// ------
void playmp3()
{
 // -----
 // Play MP3 Files (Announce Current Time)
 // -----
 if (pmp3f == 1 || pmp3f == 2)
 {
  if (!MP3player.isPlaying() && tti == 0)
  {
   MP3player.playTrack(200); //The Time is..
   delay(2000);
  }
  if (!MP3player.isPlaying() && tti == 1 && hris == 0)
  {
    hris = 1;
   MP3player.playTrack(HR);
                                  //Hour
  if (!MP3player.isPlaying() && hris == 1 && minis == 0)
  {
    if (MIN == 0)
    {
                         //O-Clock
     MP3player.playTrack(211);
     ampmis = 1;
     minis = 1;
    }
    if (MIN < 10)
    {
     MP3player.playTrack(100 + MIN);
     minis = 1;
    l
    if (MIN > 9)
    {
                                   //Minute
     MP3player.playTrack(MIN);
     minis = 1;
    }
  if (!MP3player.isPlaying() && minis == 1 && ampmis == 0)
  {
    ampmis = 1;
```

```
MP3player.playTrack(200 + AMPM);
                                 //AM or PM (200+1 or 2)
 }
 if (!MP3player.isPlaying() && ampmis == 1 && hand == 0 && pmp3f == 1)
 {
  hand = 1;
  if (good_day == 1) {
   MP3player.playTrack(210); //Have a Good Day!
  }
  else {
   MP3player.playTrack(223); //Good Night.
  }
  pmp3f = 0;
 }
}
// -----
                        _____
// Play MP3 Files (Play Alarm Track)
// ------
if (pmp3f == 3)
{
 if (!MP3player.isPlaying())
 {
 MP3player.playTrack(song); //Play Alarm Track
  pmp3f = 0;
  return;
 }
}
// -----
// STOP MP3 Files (STOP Track)
// ------
                                _____
if (pmp3f == 4)
{
 MP3player.stopTrack(); //Stop Track
pmp3f = 0;
 return;
}
```

}

Arduino Software Routines: RTC DeadOn.

Set-Up RTC:

```
_____
// Set-Up RTC
// _____
//day(1-31), month(1-12), year(0-99), hour(0-23), minute(0-59), second(0-59)
//SetTimeDate(28,10,14,11,59,50);
// ------
int RTC init()
{
 // start the SPI library:
 SPI.begin();
 SPI.setBitOrder(MSBFIRST);
 SPI.setDataMode(SPI MODE3); //mode 3 works
 //set control register
 digitalWrite(rtc_cs, 0);
 SPI.transfer(0x8E); //control reg
 SPI.transfer(0x37); //37=00100101, 60= disable Osciallator and Battery SQ wave @1hz,
                 temp compensation, Alarms disabled
 digitalWrite(rtc cs, 1);
 delay(10);
}
```

Set Time and Date on RTC:

{

```
// ------
                            _____
// Set Time and Date on RTC
// -----
// bit 7 6 5 4 3 2 1 0
// 3 2 1 0 = 0-9 BCD
//7 = 0
// 6 = 0 for 24 or 1 12 hour
//5 = 0 for AM or 1 PM
// 4 = 1 for 10 hour
// _____
int SetTimeDate(int d, int mo, int y, int h, int mi, int s)
 SPI.setBitOrder(MSBFIRST);
 SPI.setDataMode(SPI MODE3); // mode 3 works
 int TimeDate [7] = {s, mi, h, 0, d, mo, y};
 //0=second,1=minute,2=hour,3=null,4=day,5=month,6=year
 for (int i = 0; i <= 6; i++)
 {
   if (i == 3) {
    i++; //only when i=3, increment to 4.
   }
   int b = TimeDate[i] / 10;
   int a = TimeDate[i] - b * 10;
   if (i == 2)
                                           //hr
   {
     ap = AMPM;
                                           //1=am, 2=pm
     if (TimeDate[2] < 10)
     {
      if (ap == 2)
                                           //PM
       {
        TimeDate[2] = TimeDate[2] + B01100000;//bit 7=0, 6=1 (12 hr), 5=1(PM),4=0 (10hr)
                                          //bit 7 6 5 4 3 2 1 0
      if (ap == 1)
                                           //AM
        TimeDate[2] = TimeDate[2] + B01000000;//bit 7=0, 6=1 (12 hr), 5=0(AM),4=0 (10hr)
                                          //bit 7 6 5 4 3 2 1 0
       }
     }
     else if (TimeDate[2] > 9)
     {
       TimeDate[2] = TimeDate[2] - 10;
                                          //PM
       if (ap == 2)
       {
        TimeDate[2] = TimeDate[2] + B01110000;//bit 7=0, 6=1 (12 hr), 5=1(PM),4=1 (10hr)
       }
```

```
if (ap == 1)
                                              //AM
     {
       TimeDate[2] = TimeDate[2] + B01010000;//bit 7=0, 6=1 (12 hr), 5=0(AM),4=0 (10hr)
      }
   }
  }
 if (i != 2)
                                               //hr
  {
   TimeDate[i] = a + (b << 4);
  }
 digitalWrite(rtc_cs, LOW);
 SPI.transfer(i + 0x80);
                                            //write data to RTC.
 SPI.transfer(TimeDate[i]);
 digitalWrite(rtc cs, HIGH);
 delay(10);
}//for loop
```

Read Time and Date on RTC:

}

```
// -
      _____
// Read Time and Date on RTC
// ------
                     _____
void ReadTimeDate()
{
 SPI.setBitOrder(MSBFIRST);
 SPI.setDataMode(SPI MODE3); // mode 3 works
 int TimeDate [7]; //O=second, 1=minute, 2=hour, 3=null, 4=day, 5=month, 6=year
 for (int i = 0; i <= 6; i++)
 {
  if (i == 3) {
   i++;
  }
  digitalWrite(rtc cs, LOW);
                      // read RTC.
  SPI.transfer(i + 0x00);
  unsigned int n = SPI.transfer(0x00);
  digitalWrite(rtc_cs, HIGH);
                         // mask out bits 7-4.
  int a = n & B00001111;
  //-----
                     _____
  // Get Hour.
  //-----
  if (i == 2)
                         //hour
  {
   int b = (n & B00010000) >> 4; //24 hour mode

      int b10 = 10 * b;
      //= 0 or 10

      TimeDate[2] = a + b10;
      //add 0 or 10

      b = (n & B00100000) >> 5;
      //AM/PM?

      ap = 1 + b;
      //AM=1, PM=2?

  }
  //-----
  // Get Day.
  //-----
  else if (i == 4)
                        //day
  {
    int b = (n & B00110000) >> 4;
    TimeDate[i] = a + b * 10;
  //---
  // Get Month.
  //-----
  else if (i == 5) //mo
  {
   int b = (n & B00010000) >> 4;
    TimeDate[i] = a + b * 10;
  }
  //-----
  // Get Year.
  //-----
  else if (i == 6)
                         //yr
  {
    int b = (n \& B11110000) >> 4;
   TimeDate[i] = a + b * 10;
  else//sec and min
```

```
{
    int b = (n & B01110000) >> 4; //bits 6,5,4
    TimeDate[i] = a + b * 10;
  }
 }
 //-----
 // For alarming
 //-----
 RTC MON = TimeDate[5];
 RTC_DAY = TimeDate[4];
RTC_YR = TimeDate[6];
 RTC SEC = TimeDate[0];
 RTC HR = TimeDate[2];
 // Update ONLY Minutes every Minute
 if (RTC MIN != TimeDate[1] || background != background prev) // only update display at
every minute change.
 {
  background prev = background;
  RTC MIN = TimeDate[1];
  UPDATE_LCD();
 }
```

```
}
```

Example of Writing/Reading data to/from RTC SRAM:

```
// ---
      -
// Write data to RTC SRAM.
// ------
int WriteBackgroundtoRTC(int bg)
{
 // -----
 // Write background # to SRAM.
 // -----
 digitalWrite(rtc cs, LOW);
 SPI.transfer(0x098);
                         // write 5 address to address reg 98H
 SPI.transfer(5);
 digitalWrite(rtc cs, HIGH);
 delay(10);
 digitalWrite(rtc cs, LOW);
 SPI.transfer(0x99);
                        // write data to address 5 via 99h data addres
 SPI.transfer(bg);
                         // 1-4
 digitalWrite(rtc_cs, HIGH);
 delay(10);
// ------
// Read data from RTC SRAM.
// ------
void Read_RTC_SRAM()
                 // Get stored values.
{
 SPI.setBitOrder(MSBFIRST);
 SPI.setDataMode(SPI MODE3);
 // ------
 // Read Background
 // -----
 digitalWrite(rtc_cs, LOW);
                          // write 5 address to address reg 98H
 SPI.transfer(0x098);
 SPI.transfer(5);
 digitalWrite(rtc cs, HIGH);
 delay(10);
 digitalWrite(rtc_cs, LOW);
 SPI.transfer(0x19);
 background = SPI.transfer(0x05); // Get Background
 digitalWrite(rtc_cs, HIGH);
 delay(10);
 if (background < 1 || background > 4) {background = 1;}
```

Arduino Software Routines: Multi-Purpose Pushbutton.

When NOT in Alarm:

- The time will be announced when the pushbutton is pressed.
- The alarm will be enabled when the pushbutton is pressed and held for 2 seconds.

When in Alarm:

- The snooze function will activate and add 7 minutes to the alarm time and the alarm will re-trigger when the pushbutton is pressed
- The wake-up alarm will be disabled when the pushbutton is pressed and held for 2 seconds.

Set-up pushbutton I/O:

```
-
// --
// Set-Up I/O Pin Mapping and Function.
// -----
//Digital I/O Pins:
// D48 - IN alarm/snooze pb
// D49 - OUT alarm/snooze pb light
// D10 - OUT Reset the LCD
//-----
int SnoozePB = 48;
int snoozepbled = 49;
pinMode(SnoozePB, INPUT PULLUP);
pinMode(snoozepbled, OUTPUT); // PB LIGHT
// -----
// Check pushbutton status.
// -----
void Check PB Status()
// PB pressed = low
// alarm not enabled - when pressed and held for 2 seconds turn-on alarm.
// alarm not enabled - when pressed announce time.
// alarm enabled - when pressed snooze alarm enabled.
// alarm enabled - when pressed and held for 2 seconds turn-off alarm.
// -----
^{\prime\prime} alarm not enabled - when pressed and held for 2 seconds turn-on alarm.
// alarm not enabled - when pressed announce time.
// Set WU Alarm == 0 Wake up alarm disabled.
// _____
if ((digitalRead(SnoozePB) == LOW) && (PB PRESSED release == 0))
   if (Set WU Alarm == 0)
      {
        PB PRESSED release = 1;
       x = 0;
       do
       {
        delay(50);
        x = x + 1;
         if (x >= 20) {break;}
                                      // pb held down for 2 seconds.
       } while (digitalRead(SnoozePB) == LOW);
       if (digitalRead(SnoozePB) == HIGH) // Announe Time
        Snooze PB PRESSED = 3;
        Announce Time();
        return;
      }
```

```
if (digitalRead(SnoozePB) == LOW) // Turn On Alarm
      {
       Enable Wakeup Alarm();
       return;
      }
    }
// -----
// alarm enabled - when pressed snooze alarm enabled.
// alarm enabled - when pressed and held for 2 seconds turn-off alarm.
// ------
   if (Set WU Alarm == 1)
      {
       PB PRESSED release = 1;
       x = 0;
       do
       {
        delay(50);
         x = x + 1;
         if (x >= 20) {break;}
                                         // pb held down for 2 seconds.
        } while (digitalRead(SnoozePB) == LOW);
        if (digitalRead(SnoozePB) == HIGH)
                                       // Announce or Snooze
        {
         Snooze PB PRESSED = 3;
         if(!MP3player.isPlaying()) // Announce
         {
         Announce Time();
          return;
                               // Snooze
         if(MP3player.isPlaying())
         {
          Snooze Alarm();
          return;
         }
        }
        if (digitalRead(SnoozePB) == LOW) // Turn Off Alarm
        {
        Disable Wakeup Alarm();
         return;
       }
      }
 }
11
     _____
if (digitalRead(SnoozePB) == HIGH) {PB_PRESSED_release = 0;}
}
```

Alarm Clock Interface Software on PC: Software Installation.

- 1. Download files to local **c:** drive.
- 2. Open "Alarm Clock Install" folder.
- 3. Click on "setup.exe" file.

| C:\Documents and Setti | ings \A b | raham Wes | sktop\AlarmClo | ck\PC_App\ | nstall | |
|---|---------------|-----------|----------------|------------|-----------|----------|
| <u>File E</u> dit <u>V</u> iew F <u>a</u> vorites | <u>T</u> ools | Help | | | | 1 |
| 🕝 Back 🔹 🕥 🕤 🏂 | , Se | earch 😥 | Folders | | | |
| File and Folder Tasks | ۲ | | | × 🛃 | | |
| Other Places | ۲ | Support | aClock,CAB | setup.exe | SETUP.LST | |
| Details | ۲ | | | | | |

4. A "Welcome" window will appear. Click on "OK" button.

| £ | Talking Alarm Clock by ABEtronics Setup | × | | |
|------------------------|---|---|--|--|
| | Welcome to the Talking Alarm Clock by ABEtronics installation program. Setup cannot install system files or update shared files if they are in use. Before proceeding, we recommend that you close any applications you may be running. | | | |
| OK E <u>x</u> it Setup | | | | |

- 5. Install program in default directory (c:\projects\aClock)
- 6. Click on this button to start installation.

| B Talking Alarm Cock by ABEtronics Setup | × | | | |
|---|------------------|--|--|--|
| Begin the installation by clicking the button below. | | | | |
| Click this button to install Talking Alarm Clock by ABEtronics software to the specified destination directory. | | | | |
| Directory: | | | | |
| c:\projects\aClock\ | Change Directory | | | |
| | | | | |
| E <u>x</u> it Setup | | | | |

- 7. Click on "**Continue**" button to continue installation.
- 8. Set-Up completed successfully window will appear.



9. To run program, click on Start button then Talking Alarm Clock icon in list.

Alarm Clock Interface Software on PC: Default Setup File.(setupdata.ini)

• Default Setup File Settings. (setupdata.ini) Contents:

| Alarm Hour | 6 |
|-----------------|--------------|
| Alarm Minute | 00 |
| Alarm AMPM | AM |
| Alarm Track | 150 |
| Backgrounds | 1 |
| Wake Up Screens | 5 |
| Announce Hourly | 0 |
| Remote Host | 192.168.0.24 |
| Remote Port | 5000 |

<u>Note</u>: Copy "TAC App Install**aClock**" folder to "**c:\projects**" folder. This contains all files needed to run App. Overwrite existing folder.

Alarm Clock Interface Software on PC: Overview.

| Alarm Clock App ABEtronics 2018 | ver1.00 abe | X | | |
|---|--|--|--|--|
| powered by: WIZnet - Serial to Ethernet, StablE (WIZ750SR) | | | | |
| | onnection to wi2750SR! | T ABEtronics engineering services | | |
| -Date and Time Controls | – Wake Up Alarm Controls – – – – | Wake Up Screens Selection | | |
| Saturday, Jun 30, 2018 1 SET Date/Time 4:53:30 PM | Set-Up WakeUp Alarm Time 5 6 00 Al | /PM ✓ Screen 1 Hsty, | | |
| Announce Hourly 2 | Play Wakel In | n #9 Us vide sette s Full the dates | | |
| -Background/Text Set-Up | Alarm C Alarm #2 C Alarm | n #10 | | |
| ✓ Black/Green | Stop C Alarm #3 C Alarm | n #11 Screen 2 | | |
| 🗖 Black/Red 3 | Alarm C Alarm #4 C Alarm | n #12 | | |
| 🗖 Blue/Yellow | 6 C Alarm #5 C Alarm | n #13 8 RIGE UP AND PRAY | | |
| 🗖 Teal/Black | C Alarm #6 C Alarm | n #14 | | |
| | C Alarm #7 C Alarm | n #15 | | |
| -Network Communication Set-Up | C Alarm #8 C Alarm | n #16 🔽 Screen 3 | | |
| Remote Host: 192.168.0.24 | · | | | |
| Remote Port: 5000 4 | 7 Enable WakeUp Alarm | STI UP | | |
| Save and Exit | Disable WakeUp Alarm | 9 | | |

- 1: Send PC current date and time to clock.
- 2: Set announce top of the hour (toggle on/off).
- 3: Set background/text.
- 4: Set TAC IP Address and communication port.
- 5: Set wake up alarm time.
- 6: Set alarm track 1-16 (Test/Play each alarm track).
- 7: Enable/Disable wake up alarm.
- 8: Set wake up screens.
- 9: Test wake up alarm (wakeup screen, alarm track).

Note:

- Default selections are shown above.
- See demo video for more information.

X ABEtronics 2018 ver1.00 abe Alarm Clock App powered by: WIZnet - Serial to Ethernet, StablE (WIZ750SR) Connection to WIZ750SR! ABEtronics engineering services **Date and Time Controls** Wake Up Alarm Controls Wake Up Screens Selection Saturday, Jun 30, 2018 AM/PM Screen 1 Set-Up Hour Min WakeUp 00 AM. 6 Alarm Time SET Date/Time 4:53:30 PM Announce Hourly Alarm #1 C Alarm #9 Play WakeUp C Alarm #2 C Alarm #10 Background/Text Set-Up Alarm Screen 2 C Alarm #3 C Alarm #11 🔽 Black Green Stop WakeUp C Alarm #4 C Alarm #12 E Black Red Alarm C Alarm #5 C Alarm #13 Blue/Yellow KISE UP AND O Alarm #6 O Alarm #14 Teal Black C Alarm #7 C Alarm #15 Screen 3 C Alarm #8 C Alarm #16 Network Communication Set-Up WAKE Remote Host: 192.168.0.24 Enable WakeUp Alarm ΠÞ Remote Port: 5000 TEST! Save and Exit Disable WakeUp Alarm

Alarm Clock Interface Software on PC: Set Current Date/Time.

Set Current Time:

- 1. Click on "**Set Date/Time**" button to send the current PC's date and time to the TAC.
- 2. The TAC will start updating the current time.



Note:

- When Times are sent to the TAC, the downloaded times will be announced to insure the correct time was downloaded.
- All data is stored in the RTC battery backed up RAM.

Alarm Clock Interface Software on PC: Announce Time at the Top of the Hour

| Alarm Clock App ABEtronics 2018 | ver1.00 abe | × | | |
|---|--|---|--|--|
| powered by: WIZnet - Serial to Ethernet, StablE (WIZ750SR) | | | | |
| о с | onnection to WIZ750SR! | ABEtronics engineering services | | |
| Date and Time Controls | Wake Up Alarm Controls | Wake Up Screens Selection | | |
| Saturday, Jun 30, 2018 SET Date/Time 4:53:30 PM | Set-Up WakeUp Alarm Time 6 00 AM | Creen 1 | | |
| Announce Hourly Background/Text Set-Up | Play C Alarm #1 C Alarm #9 WakeUp Alarm C Alarm #2 C Alarm #10 | Use visite carrière is Fill Int No dilings | | |
| ✓ Black/Green ✓ Black/Red | Stop WakeUp Alarm C Alarm #3 C Alarm #11 C Alarm #4 C Alarm #12 | Screen 2 | | |
| ☐ Blue/Yellow ☐ Teal/Black | C Alarm #5 C Alarm #13 C Alarm #6 C Alarm #14 | RISE UP AND PRAY | | |
| Network Communication Set-Up | C Alarm #7 C Alarm #15 C Alarm #8 C Alarm #16 | | | |
| Remote Port: 5000 | Enable WakeUp Alarm TEST! | UP | | |
| Save and Exit | Disable WakeUp Alarm | | | |

Announce Time at the Top of the Hour:

Click on "**Announce Hourly**" check box select this option. Your option will be downloaded to the TAC.

At the top of the hour the TAC will display the time and the time will be announced.



Click on "**Announce Hourly**" check box again to de-select this option. Your option will be downloaded to the TAC.

Note: (whenever the time is announced)

Depending on the time of day and after the time is announced will say "Good Morning", "Good Afternoon", "Good Evening" or "Good Night".

Alarm Clock Interface Software on PC: Select Background Screens.

| Alarm Clock App ABEtronics 2018 | ver1.00 abe | × | | |
|--|------------------------------|---|--|--|
| powered by: WIZnet - Serial to Ethernet, StablE (WIZ750SR) | | | | |
| o c | onnection to WIZ750SR! | ABEtronics engineering services | | |
| Date and Time Controls | Wake Up Alarm Controls | Wake Up Screens Selection | | |
| Saturday, Jun 30, 2018 | Set-Up Hour Min AM/PM | Screen 1 | | |
| SET Date/Time 4:53:30 PM | Alarm Time 6 00 AM | Holy, Holy, Holy is not LCRD Almighty: | | |
| Announce Hourly | Play Watel In | be vide on to | | |
| Background/Text Set-Up | Alarm C Alarm #2 C Alarm #10 | and both | | |
| ✓ Black/Green | Stop C Alarm #3 C Alarm #11 | Screen 2 | | |
| ☐ Black/Red | Alarm C Alarm #4 C Alarm #12 | | | |
| ☐ Blue/Yellow | C Alarm #5 C Alarm #13 | RISE UP AND PRAY | | |
| 🗖 Teal/Black | C Alarm #6 C Alarm #14 | All all and a second | | |
| | ⊖ Alarm #7 ⊖ Alarm #15 | | | |
| Network Communication Set-Up | C Alarm #8 C Alarm #16 | Creen 3 | | |
| Remote Host: 192.168.0.24 | | WAKE | | |
| Remote Port: 5000 | Enable WakeUp Alarm TEST! | UP | | |
| Save and Exit | Disable WakeUp Alarm | | | |

Select Background and Text:

- 1. Click on **desired check box**.
- 2. Your selection will be downloaded and displayed on the TAC.





Alarm Clock Interface Software on PC: Set TAC IP Address and communication port.



Enter the WIZ750SR Module IP Address:

- 1. Click on "**Remote Host**" entry field to set the IP address found on the WIZ750SR configuration tool on page 6.
- 2. Click on "**Remote Port**" entry field to set the remote port number found on the WIZ750SR configuration tool on page 6. Default is 5000.
- 3. Click on "Save and Exit" button to save network configuration.
- 4. Start the PC app again and now will connect to WIZ750SR module.

Note:

When the PC app is communicating with the WIZ750SR module the indicator will show yellow. If not communicating will show red and will need to restart the app.



Alarm Clock Interface Software on PC: Set Wake Up Alarm Time.

Set-Up Alarm Time:

- 1. Click on "Hour/Min/AM/PM" entry field to set the desired alarm time.
- 2. Click on "**Set WakeUp Alarm/Time**" button to send the desired alarm time to the TAC. Will also download the alarm track selected.
- 3. The TAC will display the alarm time.

Click on the "Enable Wakeup Alarm" button to enable wake alarm. Box on screen will turn yellow.

Note:

The alarm can also be enabled when the pushbutton is pressed and held for 2 seconds.



Note:

- When Times are sent to the TAC, the downloaded times will be announced to insure the correct time was downloaded.
- All data is stored in the RTC battery backed up RAM.

Alarm Clock Interface Software on PC: Set Wake-Up Alarm & Screens.

| Alarm Clock App ABEtronics 2018 | ver1.00 abe | X | | |
|--|------------------------------|--|--|--|
| powered by: WIZnet - Serial to Ethernet, StablE (WIZ750SR) | | | | |
| Connection to WIZ750SR! | | ABEtronics engineering services | | |
| Date and Time Controls | Wake Up Alarm Controls | Wake Up Screens Selection | | |
| Saturday, Jun 30, 2018 | Set-Up Hour Min AM/PM | 🖌 🔽 Screen 1 | | |
| SET Date/Time 4:53:30 PM | Alarm Time 6 00 AM | Holys Holys Holys is the LCRD Almighty: | | |
| Announce Hourly | Play Wakel In | Use visite surfic si Falloni Hologo | | |
| Background/Text Set-Up | Alarm C Alarm #2 C Alarm #10 | | | |
| ✓ Black/Green | Stop C Alarm #3 C Alarm #11 | Screen 2 | | |
| ☐ Black/Red | Alarm C Alarm #4 C Alarm #12 | | | |
| ☐ Blue/Yellow | C Alarm #5 C Alarm #13 | RISE UP AND PRAY | | |
| Teal/Black | C Alarm #6 C Alarm #14 | 128 Bar | | |
| | ⊂ Alarm #7 ⊂ Alarm #15 | | | |
| Network Communication Set-Up | C Alarm #8 C Alarm #16 | Screen 3 | | |
| Remote Host: 192.168.0.24 | | | | |
| Remote Port: 5000 | Enable WakeUp Alarm TEST | UP | | |
| Save and Exit | Disable WakeUp Alarm | | | |
| | | | | |

Set Wake-Up Alarm:

- To Set Alarm: Click on "Enable WakeUp Alarm".
- To Reset Alarm: Click on "Disable WakeUp Alarm".

Select Wake Up Screens:

- 1. Click on "option button 1, 2 or 3".
- 2. Your selection will be downloaded and displayed on the TAC.
- 3. Click on "Disable WakeUp Alarm" to return the main clock screen.



Alarm Clock Interface Software on PC: Play Alarm Tracks.

Play Alarm Tracks:

- 1. Click on desired alarm # 1-16 option box.
- 2. Click on "Play WakeUp Alarm" button to hear your selection on the TAC.
- 3. Click on "Stop WakeUp Alarm" button to stop the alarm track.

Note:

To use as a wake up alarm, click on the "Set-Up WakeUp Alarm Time" button.