Infrared Remote Controller Workshop
User's Guide

Revision: V1.00 Date: June 03, 2021

www.holtek.com

Table of Contents

1. Introduction

The HOLTEK Infrared Remote Controller Workshop is a design platform for the fast design of infrared remote controllers. The software not only supports remote controller software development based on general standard protocols, such as NEC, NEC-16, Philips RC-5, Philips RC-6, Sharp and other common remote controller protocols, but also provides a design method for users to define their own infrared remote code parameters which are different from other standard protocols. In addition, together with the HOLTEK IR decoder board, the workshop can analyse IR waveforms for existing remote controllers. This can then be used for new remote controller development based on the same protocol or used to verify the sending code for remote controllers under development.

The main application functions for common infrared remote controllers are divided into the following categories:

1. General remote controllers

   Keys: identify user actions.

   Code modulation: use different code protocols to modulate signals to avoid interference between different receiver devices.

   Indicators: code indication.

   Static power control: extend battery life.

2. LCD remote controllers

   Keys: identify user actions.

   Code modulation: use different code protocols to modulate signals to avoid interference between different receiver devices.

   LCD display: display various operation status information that the remote controller sends to the receiver device.

   Backlight: LCD backlight driver.

   Static power control: extend battery life.

2. Workshop Software

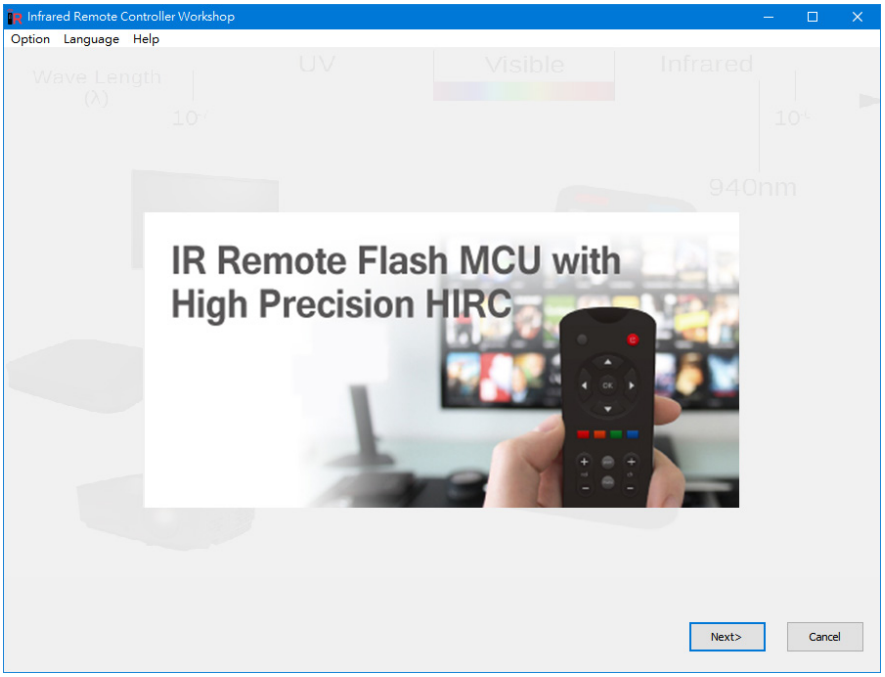


Fig.1 Boot Screen

After the software has started as shown in Fig.1, the menu bar at the top left will show the following menus from left to right: Option, Language and Help.

The option menu provides an update F/W and a reset functional configuration demonstration functions.

The language menu provides a choice of Simplified Chinese, Traditional Chinese and English.

The help menu provides a user's guide, MCU datasheet, development board schematic design, infrared remote controller knowledge base and workshop version information.

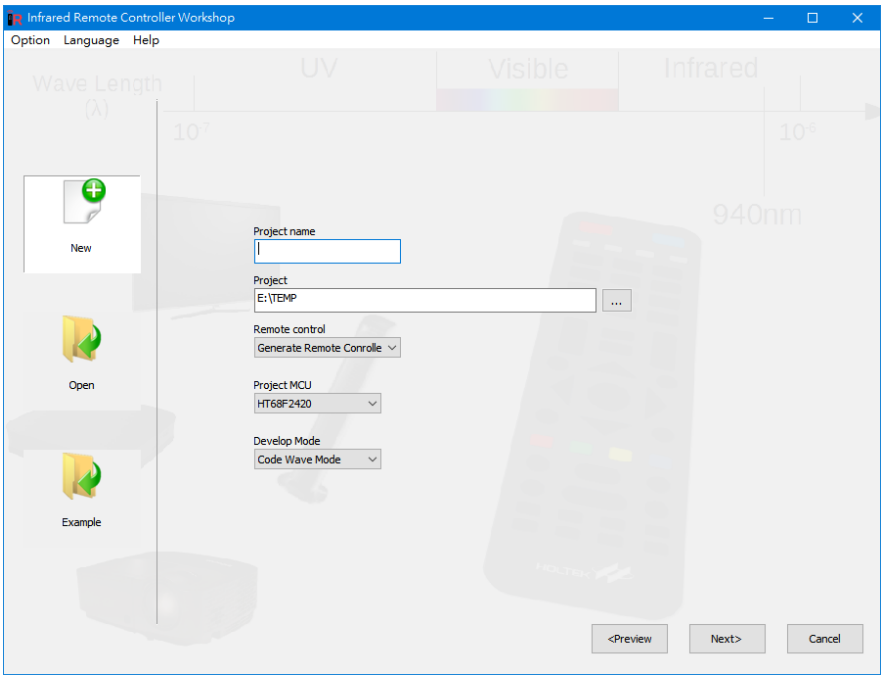


Fig.2 Project Setting

In the project setting view, users can create a new project, open an old project or an existing example. A new remote controller development project can be implemented by selecting either the general remote controller or LCD remote controller in the new project setting. For the general remote controller development, the supported MCU is the HT68F2420 and the development mode can choose the standard protocol mode, the user defined protocol mode and the code mode. For the LCD remote controller development, the supported MCU is the HT67F2432 and the development mode can choose either the user defined protocol mode or the code mode.

The remote controller development process in the workshop software consists of the following main configuration steps:

1. Protocol definition
2. LCD display content – for LCD remote controllers only
3. Key and driver configurations
4. Generate a project which can be used for secondary development and then compile it to generate the programming files

2.1 General Remote Controller Development

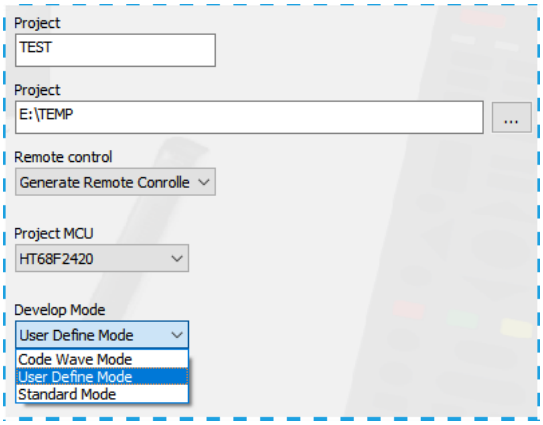


Fig.3 Project Setting for General Remote Controllers

The general remote controller development supports three development modes: standard protocol mode, user defined protocol mode and code mode. In the project setting page, select the general remote controller type. Here the supported MCU is the HT68F2420.

2.1.1 Standard Protocol Mode Development

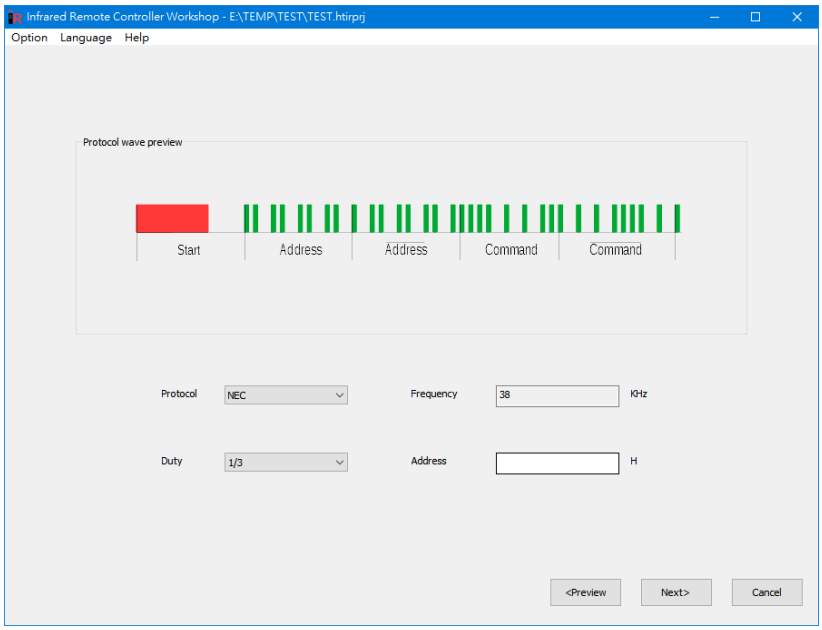


Fig.4 Standard Protocol Mode Configuration

As shown in Fig.4, in the standard protocol development mode, NEC, Philips RC-5, Philips RC-6, Sharp, JVC and other popular protocols in the market can be used for development. When using a standard protocol, it is not required to configure any protocol controlled parameters. In addition, the software view provides a waveform preview. After the standard protocol which is to be used is selected, it is only necessary to configure the appropriate duty and address code to complete the configuration of the protocol part.

2.1.2 User Defined Protocol Mode Development

The user defined protocol development consists of two tabs to configure the parameters. The first is the modulation mode tab. In this tab, the carrier parameters, Bit1 and Bit0 modulation parameters, boot code parameters and repeat code parameters can be defined by users.

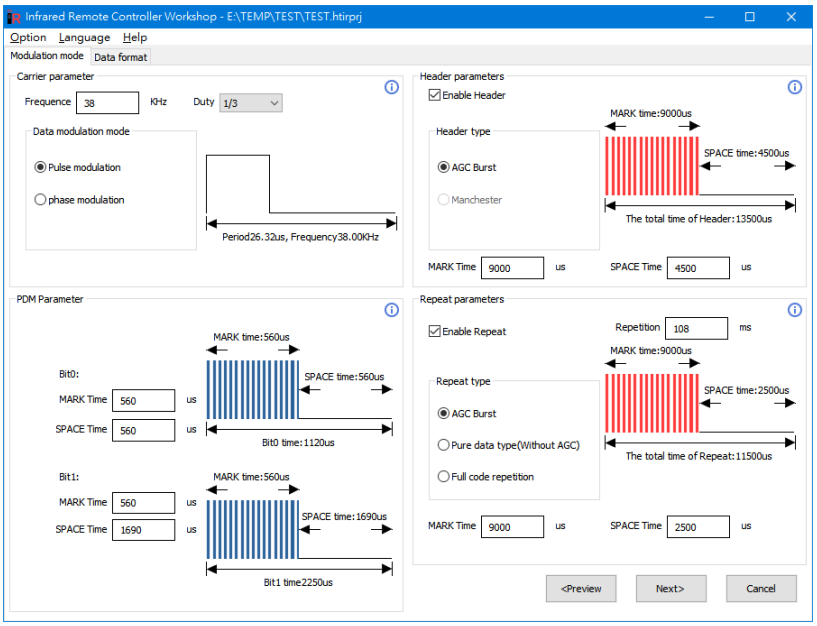


Fig.5 User Defined Protocol Mode Configuration

The specific configuration operations are as follows:

Carrier

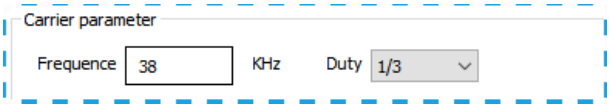


Fig.6 Carrier Parameters

As shown in Fig.6, in the user defined protocol mode, the carrier parameters can be user defined. The carrier frequency ranges from 30kHz to 58kHz and the duty can select 1/2, 1/3 or 1/4.

Data Modulation Mode

The data modulation mode can select either pulse modulation (PDM) or phase modulation (Manchester).

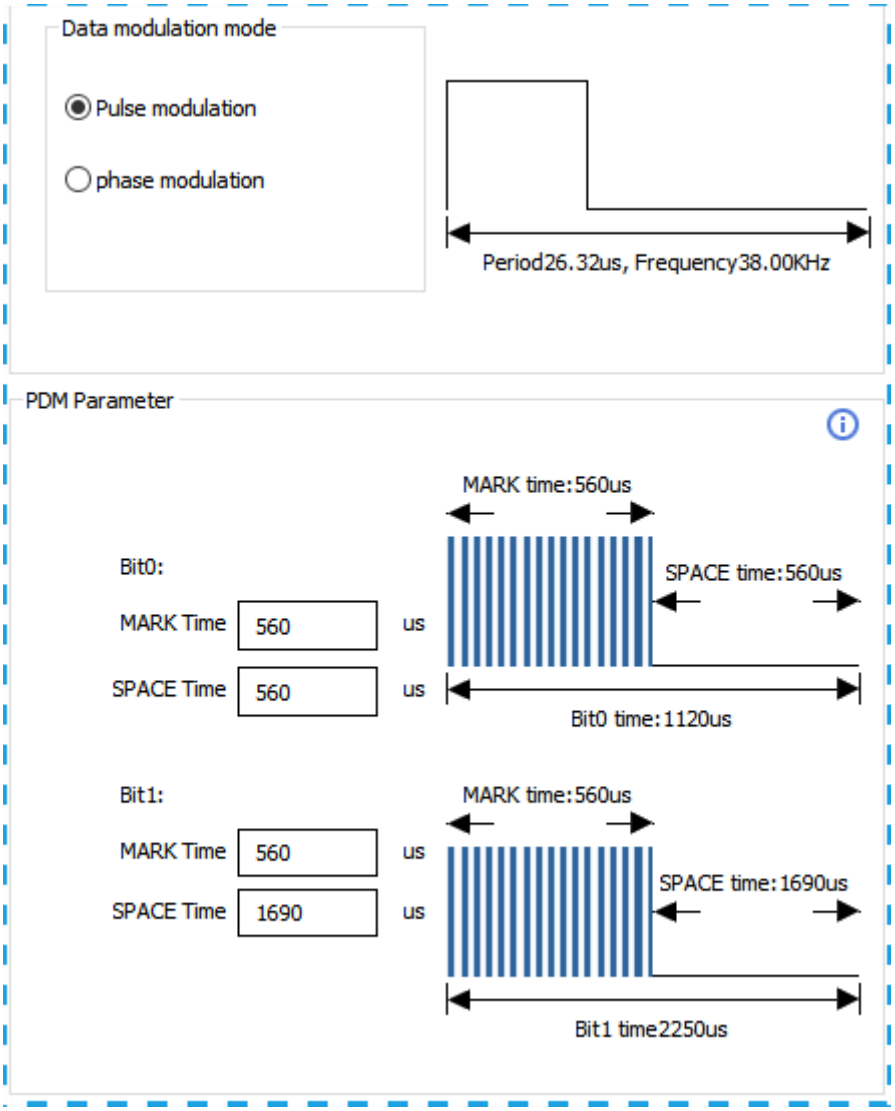


Fig.7 Pulse Duration Modulation (PDM) Parameters

Fig.7 is a configuration diagram showing the pulse duration modulation parameters of Bit0 and Bit1. In the software, the MARK (Continuous Carrier Pulse) time and SPACE (No Pulse) time for Bit0 and Bit1 in the modulation signal sent by the remote controller can be configured. The MARK, SPACE and total time parameters are marked graphically on the right side by the software.

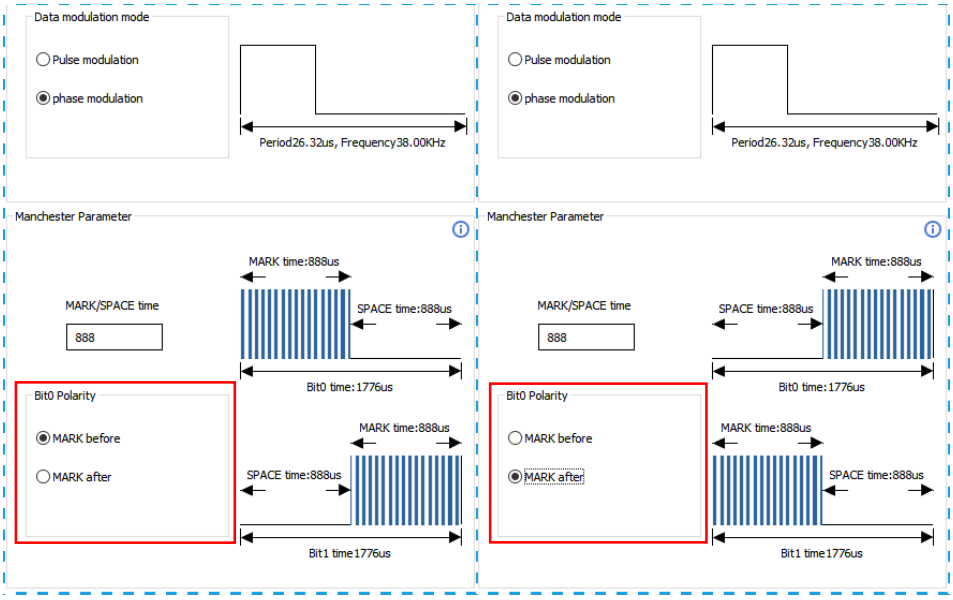


Fig.8 Phase Modulation (Manchester) Parameters

Fig.8 is a configuration diagram showing the phase modulation parameters. In the software, the MARK/SPACE time of Bit0 and Bit1 can be configured. For the phase modulation, only one time parameter needs to be set. Then use polarity definition to select whether the MARK time of Bit0 is in front or behind. The MARK, SPACE and total time parameters are marked graphically on the right side by the software.

The MARK and SPACE parameters can be configured within a range of 300μs~2000μs.

Boot Code/Header Parameters

The Boot Code, also known as a Header, is used by the remote controller to send an initial calibration signal before sending the control signals to a receiver.

The boot code supports both AGC Burst and Manchester types. The Manchester (phase modulation) type boot code is only available when the data phase modulation is selected.

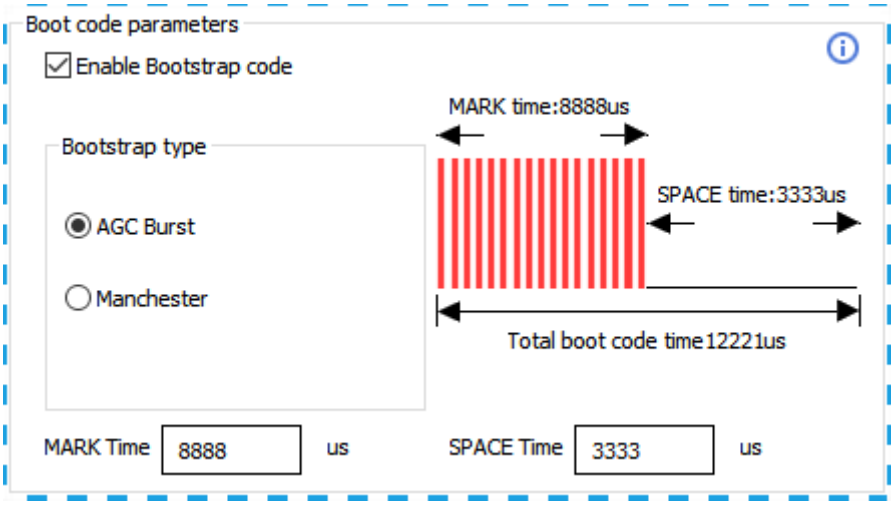


Fig.9 AGC Burst Type Boot Code Parameters

As shown in Fig.9, the AGC Burst type boot code consists of a period of MARK and SPACE that is longer than the general data code. The AGC Burst type boot code assists the receiver to initialise the gain, but the MARK time is proportional to the power consumption. Therefore this parameter should be configured correctly. It is recommended to set this between 4ms and 20ms. The MARK, SPACE and total time parameters are marked graphically by the software.

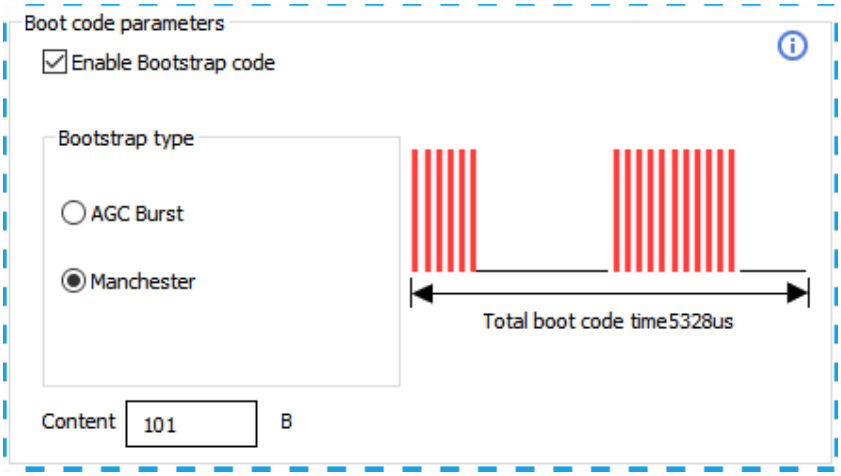


Fig.10 Manchester Type Boot Code Parameters

As shown in Fig.10, when data is modulated using the Manchester mode, a digital boot code similar to the Philips RC-5 protocol can be used. The boot code in the software can be configured with up to 3 bits. The positions of MARK and SPACE as well as total time parameter are marked graphically by the software.

Repeat Code Parameters

The Repeat Code is used for the remote controller to send a repeat signal when a button is held down. As shown in Fig.11, when the repeat code is enabled, the repeat code parameters can be configured. The repeat code is categorised into AGC Burst, pure data repeat and full code repeat types.

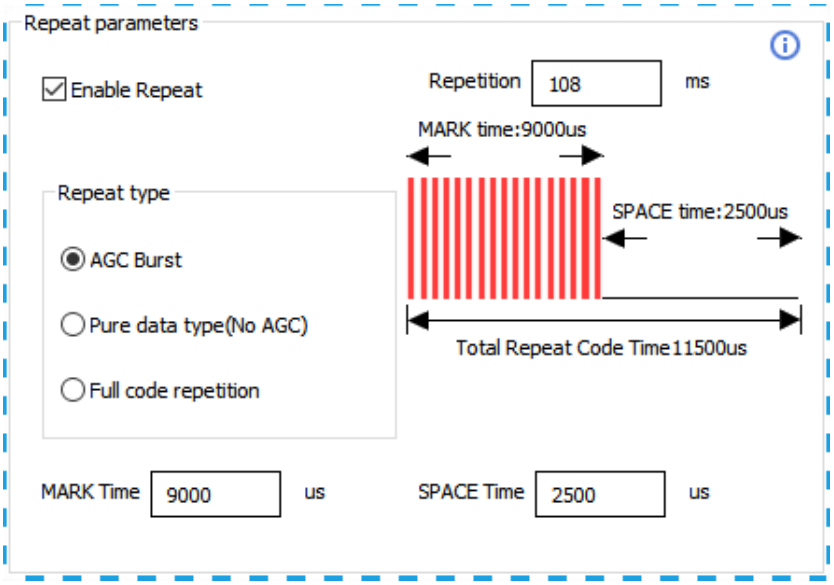


Fig.11 Repeat Code Parameters

Only the AGC Burst type repeat code needs to be configured, which can be inconsistent with the header. It is recommended to set this between 4ms and 20ms.

To ensure that the repeated second frame signal is sent 10ms later after the end of the first frame signal, the repetition period can be set between 40ms and 200ms.

Data Format

As shown in Fig.12, in the second tab, the data format parameters can be set, such as the number of address bits, the number of command bits, the sending order and whether to insert Start and Toggle bits or not.

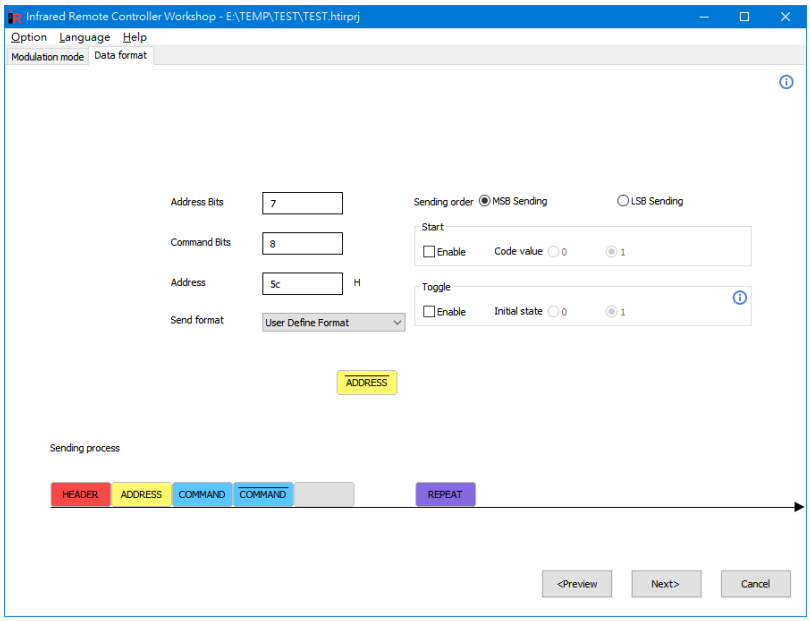


Fig.12 Data Format Parameters

Address bits: The number of bits is in a range from 1 to 8.

Command bits: The number of bits is in a range from 1 to 8.

Address: The maximum value of the address code depends on the number of address bits. The address code is entered in hexadecimal format.

Sending order: Define how data is sent, either MSB or LSB first.

Start: A fixed bit is inserted in front of the address and command code.

Toggle: A variable bit is inserted to distinguish whether the two codes sent by a remote controller in the full code repeat mode is triggered by a long key press condition or by a two consecutive press conditions.

Sending process: The sending process is shown with a timeline graph. The boot code is sent first, then the address code, the complement of the address code, the command code and the complement of the command code, can all be chosen by dragging. The sending order can be adjusted arbitrarily. When the Start or Toggle bit is enabled, these two bits are placed behind the Header and in front of the Address and Command.

2.1.3 Step Matrix Keys and Driver

When the standard protocol or user defined protocol parameters have been configured, the software will jump to the next configuration page which is for the matrix keys. As shown in the following figure, area A is an I/O area for selection, area B is a supported MCU package prompt area for the current

configuration scheme, area C is an IR LED driver mode selection area, area D is an I/O configured key area, area E is a command code write area and area F is a saved key area for code development.

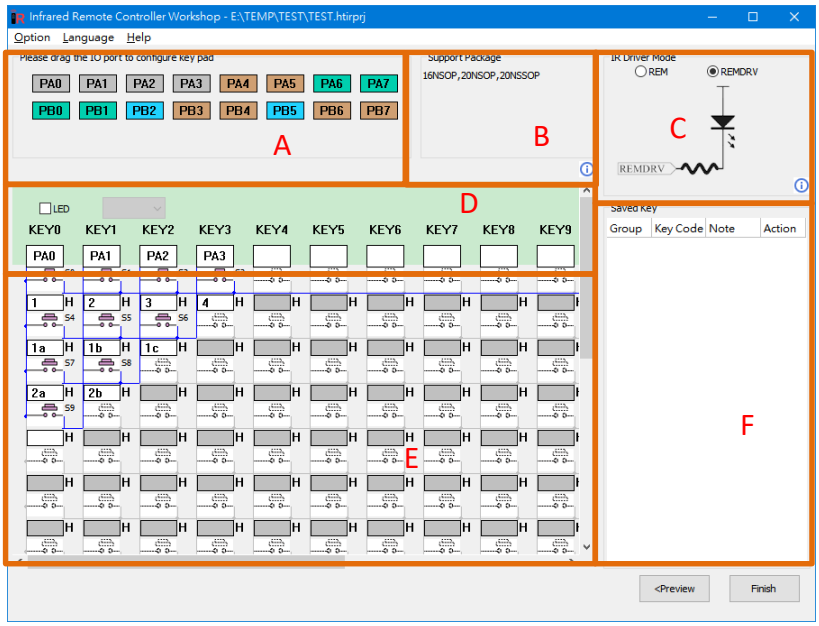


Fig.13 Key Configuration Page

2.1.3.1 Keys and I/Os

Area A, at the top left of the configuration page, shows the available I/O pins. Here the blue pin icon is common to all package types of the device, the yellow pin icon is unique to the 16NSOP and 20SSOP/20NSOP packages and the green pin icon is unique to the 20SSOP/20NSOP packages. The I/O configuration can be implemented by dragging an I/O icon into the key position below area D. Area B indicates the available package types according to the current key configuration.

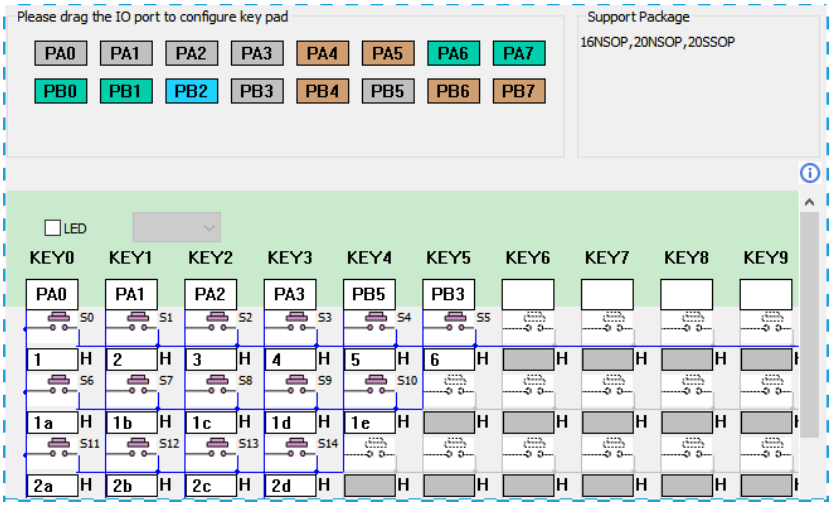


Fig.14 Key Configuration Diagram

As shown in Fig.14, when the I/O pins are configured to the Key configuration table, a step matrix keyboard is generated automatically. The maximum number of keys that can be generated is calculated using the formula: K = 1+2+…+N, where N is the number of the configured I/O pins. The

8-pin package can configure up to 15 keys. The 16-pin package can configure up to 78 keys. The 20-pin package can configure up to 136 keys.

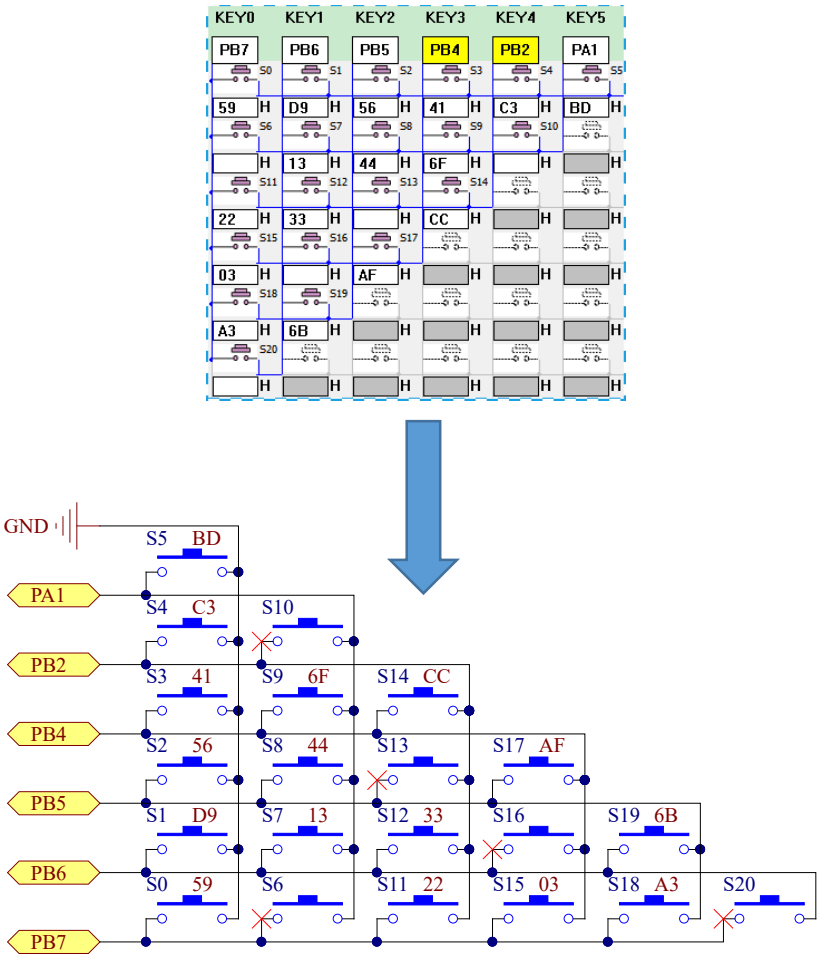


Fig.15 Relationship between Key Configuration and Schematic Diagram

After the I/O pins in the software are configured to the key area, the command code can be entered into the corresponding key code table. When clicking on any configurable key code, two I/O pins corresponding to this command code will be highlighted in yellow. In the example shown in the Fig.15 upper half, PB4 and PB2 are the two I/Os connected with key S14.

When clicking on the command code on the first line, only one I/O will be highlighted in yellow. This is because the keys corresponding to this code consists of I/Os and VSS.

Fig.15 is a step keyboard schematic diagram corresponding to the step matrix key table below. Keys without configured key values are represented by crosses in the schematic diagram. S6, S10, S13, S16 and S20 in the figure have been included in the scanned code table during configuration generation, even if the key values have not been configured or if the keys have not been soldered onto the physical circuit. For unconfigured keys, fill in 00H in the software-generated key code table.

2.1.3.2 LED Driver

The LED driver includes an MCU dedicated to drive an IR LED transmitter and an I/O to drive an LED indicator.

The IR LED driver mode is shown in Fig.16 below.

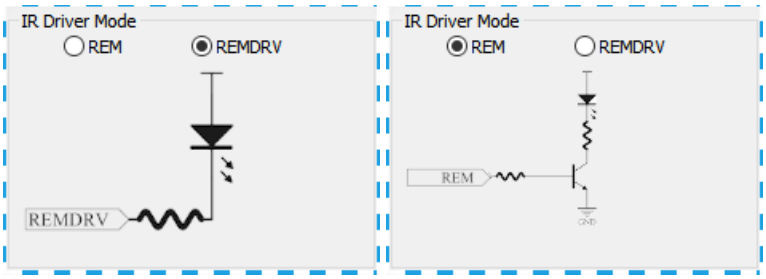


Fig.16 IR LED Driver Configuration

The IR LED that transmits the infrared signal is driven by the remote controller via the MCU REM or REMDRV pin. The REMDRV driver mode can provide up to 500mA of drive current at 3V. If a greater drive capacity is required, the REM driver mode combined with an external bipolar or MOSFET transistor can be used to improve the output drive capability.

The I/O pin to drive an LED indicator can be selected in area D, either for a functionally independent pin or for a pin-shared function I/O pin. However, the internal step keyboard scanning program will check all I/O pins using a low level except for KEY0. Therefore, if the LED indicator drive pin is pin-shared with a key pin, in order to reduce power consumption and prevent the indicator from being illuminated, the LED indicator must be configured at KEY0.

Fig.17 Indicator Configuration

After the key configuration is complete, click finish to generate the project files of the development scheme.

The automatic naming rule for the generated files is "project name_year-month-day-hour-minute-second".

2.2 LCD Remote Controller Development

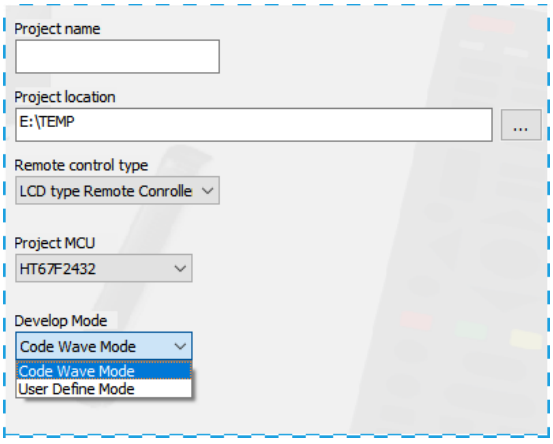


Fig.18 Project Setting for LCD Remote Controllers

The LCD remote controller development supports two development modes: user defined protocol mode and code mode. In the project setting page, select the LCD remote controller type, here the supported MCU is the HT68F2432.

The LCD remote controllers are most widely used in air conditioner remote controller products. At present the software configures and develops the remote controllers according to the air conditioner

functions. In the future, updated versions of the software will increase the range of remote controller product development types.

2.2.1 Protocol Definition

The workshop software provides a user defined protocol mode for LCD remote controller development. Users can define their own parameter and code rules according to their product requirements.

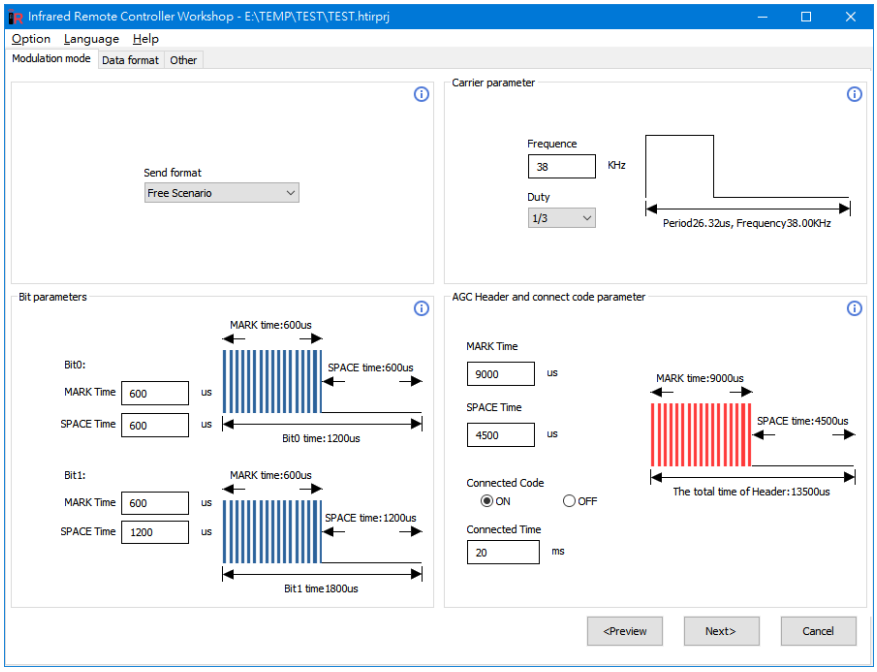


Fig.19 Modulation Mode Configuration Page

2.2.1.1 Modulation Mode

The first step in configuring the code is to configure the modulation mode. The LCD remote controller code can only use the PDM modulation mode and does not support the phase modulation mode. The modulation mode page contains four parts, which are send format, carrier parameters, Bit parameters, header (boot code) and connect code parameters.

- Send Format

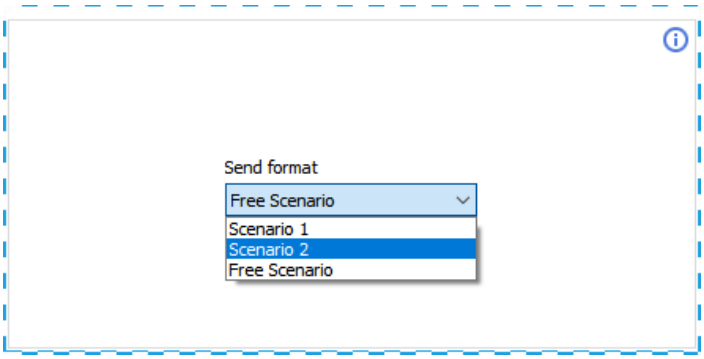


Fig.20 Send Format

In the send format, the software provides two integrated design schemes which are configured with different modulation modes, modulation parameters and data formats.

- Carrier Parameters

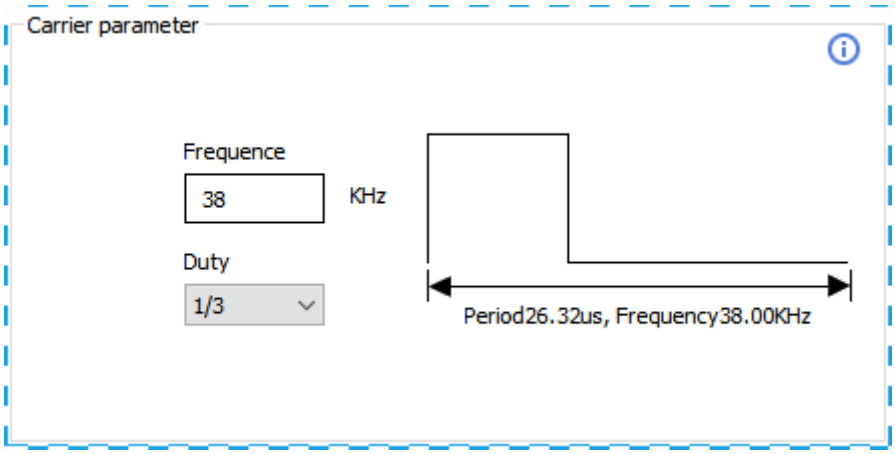


Fig.21 Carrier Parameters

The carrier parameters can be set by users, such as frequency and duty. The carrier frequency ranges from 30kHz to 58kHz and the duty can select 1/2, 1/3 or 1/4.

- Bit Parameters

In the Bit Parameters area, the MARK (Continuous Carrier Pulse) and SPACE (No Pulse) time of Bit0 and Bit1 can be configured. The MARK and SPACE time parameters are marked graphically by the software. The MARK and SPACE time can be set to between 300μs and 2000μs.

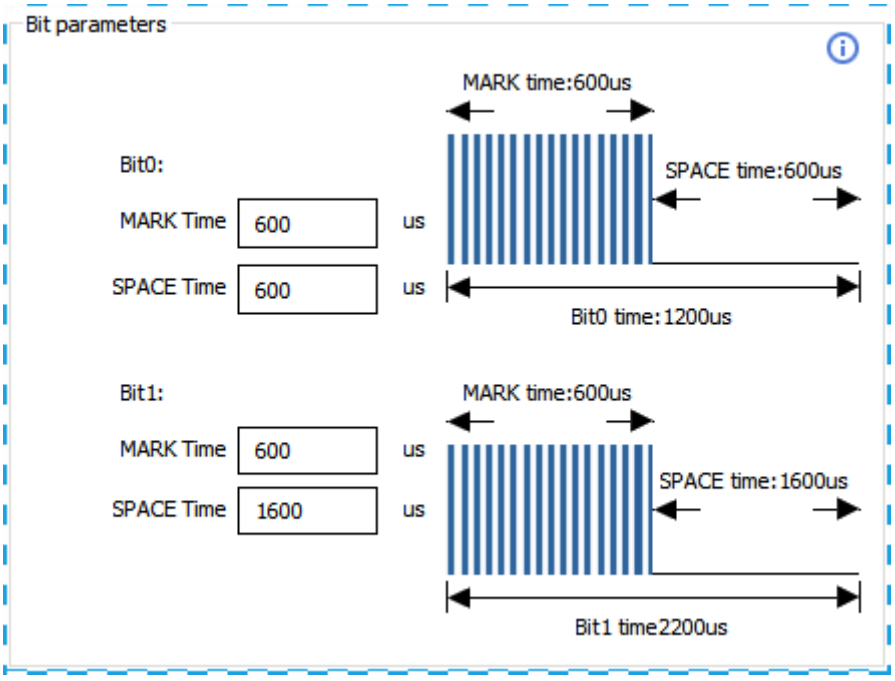


Fig.22 Bit Parameters

- Header and Connect Code Parameters

  The Header, also known as the Boot Code, is used by the remote controller to send an initial calibration signal to a receiver. The boot code for LCD remote controllers only supports the AGC Burst type. The AGC Burst type boot code consists of a period of MARK and SPACE that is longer than the general data code. The longer MARK in the AGC Burst type boot code assists the receiver to initialise the gain, but the MARK time is proportional to the power consumption. Therefore this parameter should be configured correctly to ensure that the battery life can be extended. It is recommended to set this to between 4ms and 20ms. The MARK, SPACE and total time parameters are marked graphically by the software.

  The Connect Code is used to insert a long SPACE in the middle of a long string of bit codes in the LCD remote controller. The connect code can not only be used for the decoding software to identify the protocol but can also be used to avoid the MCU LVR reset problem due to rapid voltage drops caused by continuous code transmission when the battery capacity is low.

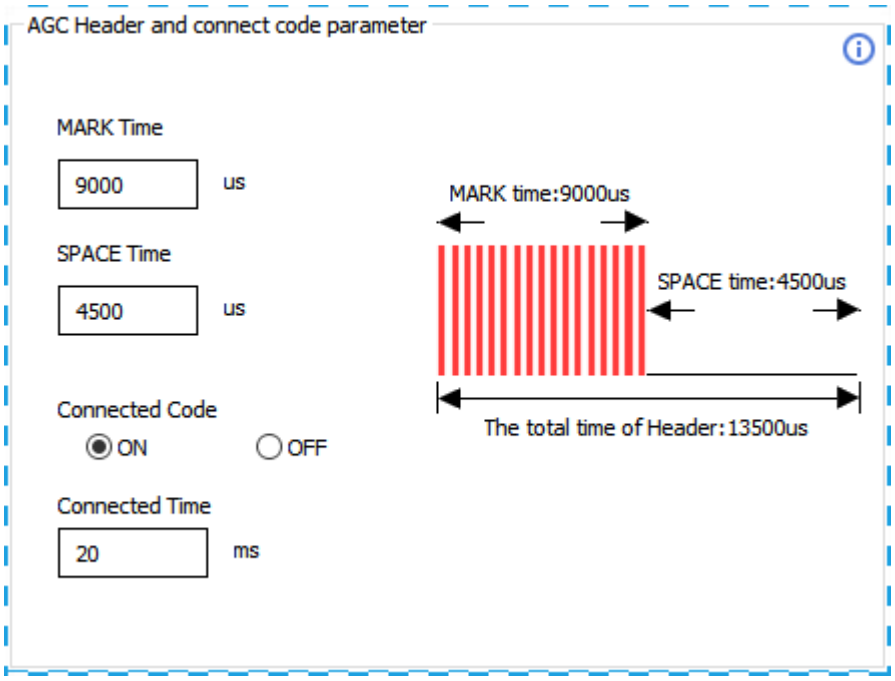


Fig.23 Header and Connect Code Parameters

2.2.1.2 Data Format

The second step in configuring the code is to configure the data format. Most of the protocols used by the LCD remote controller modulate the LCD display content into the sending data, which varies with the display content for various LCD remote controller products. Refer to the infrared remote controller knowledge base description in the software for more details.

The most widely used LCD remote controllers in consumer products are for air conditioner remote controllers. The user defined data configuration in the HT67F2432 is designed according to the common functions for air conditioner remote controllers. As more MCUs are added, more allowed functions and configurations will be included.

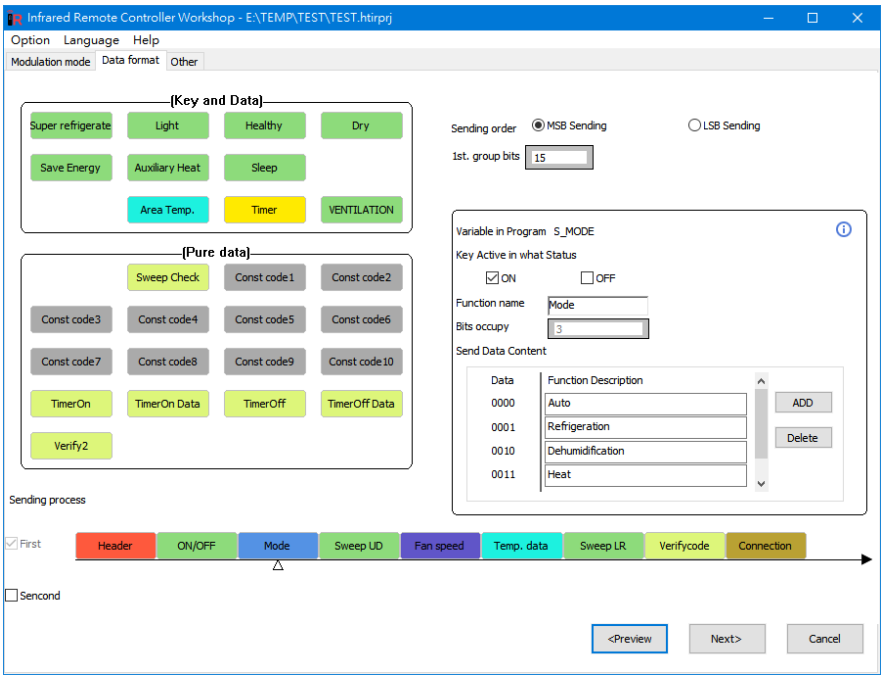


Fig.24 Data Format Page

The data format page is mainly used to configure the functional data. At the top right corner, the code sending order can be set, either LSB or MSB first.

At the bottom the sending process is shown with a timeline graph. The boot code is to be sent first, then the mode, on/off, temperature data and fan speed are fixedly used in the sending process.

In the left function area, various optional block icons are designed according to different functions. When the remote controller requires to send any function data, it can be dragged to the sending process below. Data with key, at the upper left, is a set of sending data with the key function. Pure data, at the lower left, is a set of fixed data or calculated data resulting from other key operations.

When clicking on any set of data, the data functions can be configured in the right data configuration area.

The data parameters shown on the right side are as follows:

Variable in program: the variable in the generated project corresponding to the currently configured data;

Key active status: select the LCD display status in which the currently configured key can respond to key actions, either the power-on, power-off or both states;

Function name: the default name of the current configuration;

Bits occupy: the number of bits for the current configuration occupied in the sending process. The default data with key cannot be changed during configuration (can be modified during secondary development). For pure data the number of bits can be defined by users during configuration.

Mode selection: Certain functions that do not require key response in all modes can be designed in this configuration to be available in specific modes. The selectable modes depend on the enabled modes in the mode configuration;

Sending data content design: The sending data content for general functions is configured as 0 and 1. The mode, fan speed and area temperature function data can be configured with multiple modes.

- On/Off

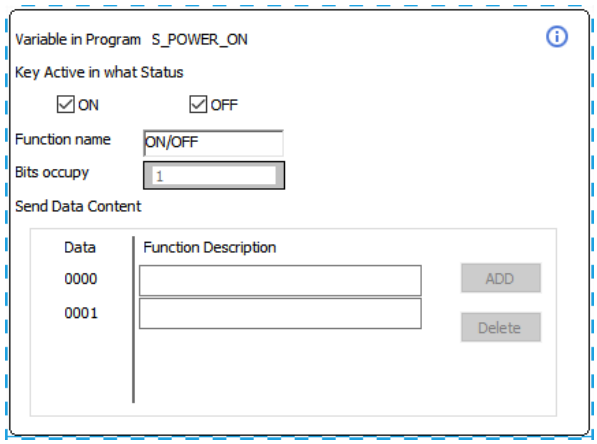


Fig.25 On/Off Key Configuration

The on/off function should be configured to execute key actions in both the power-on and power-off states. The default definition for the generated program is 1 for on and 0 for off.

- Mode

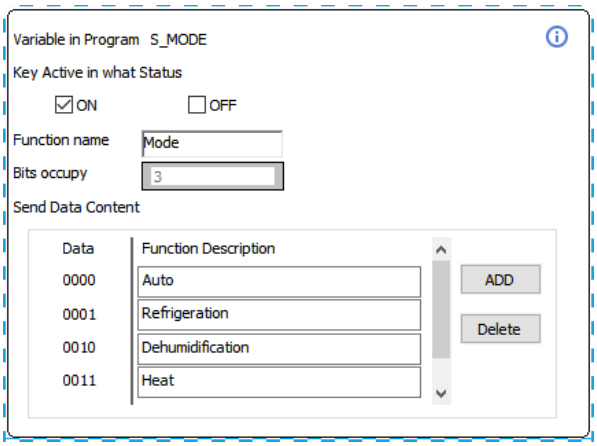


Fig.26 Switch Type Mode Key Configuration

The mode function is generally configured to execute key actions in the power-on state. In the sending data content design, the operating modes can be configured using the Add and Delete buttons, with up to 8 modes. The functional description can be written for each mode in the software. The LCD display for the mode function is to switch the segment to display.

- Sweep Up and Down, Sweep Left and Right, Healthy, Ventilation, Light

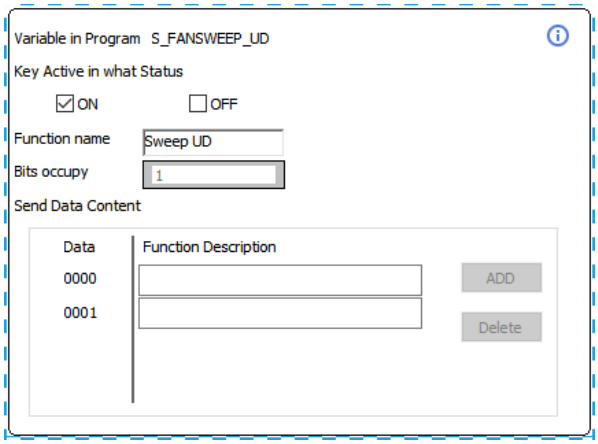


Fig.27 On/Off Type Key Configuration

These functions are configured in a similar way, generally to execute key actions in the power-on state. The default definition for the generated program is 1 for on and 0 for off. The LCD display for the sweep up and down, sweep left and right, healthy, ventilation and light functions is to turn on the segment to display.

- Fan Speed

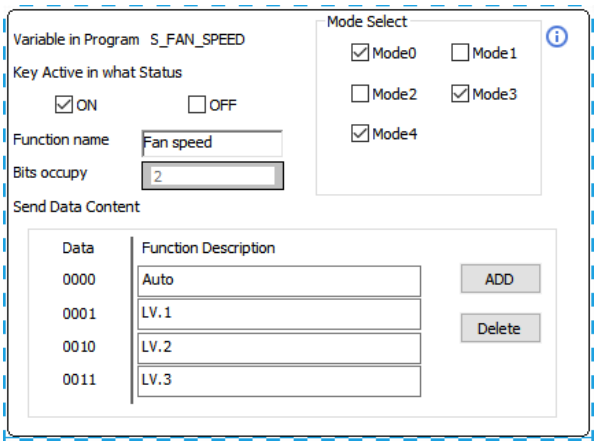


Fig.28 Upshift Type Key Configuration

The fan speed function is generally configured to execute key actions in the power-on state. In the sending data content design, various fan speed levels can be added using the Add button, with up to 8 speed modes. The default level 0 is the auto mode. In the operating mode at the top right, the specified mode supports shift actions. When the fan speed function is combined with an LCD display, Level 0 as an auto mode is a switching display, other levels are an incremental display.

- Super Refrigerate, Save Energy, Sleep, Auxiliary Heat, Dry

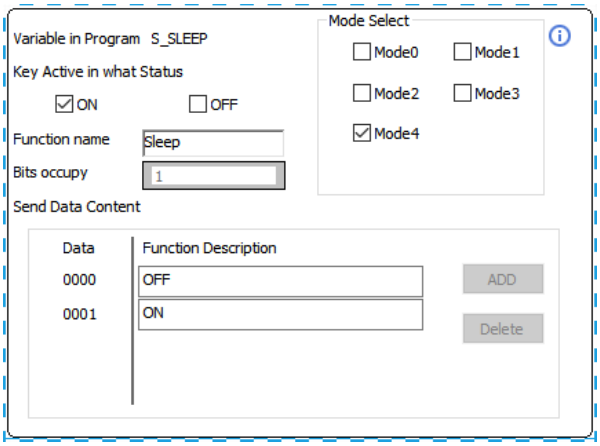


Fig.29 On/Off Type Key Configuration for Specified Operating Modes

These functions are configured in a similar way, generally to execute key actions in the power-on state. The default definition for the generated program is 1 for on and 0 for off. In the upper right operating mode, users can choose which mode to support the shift actions. The LCD display for the super refrigerate, energy saving, sleep, auxiliary heat and dry functions is to turn on or off the segment to display.

- Area Temperature

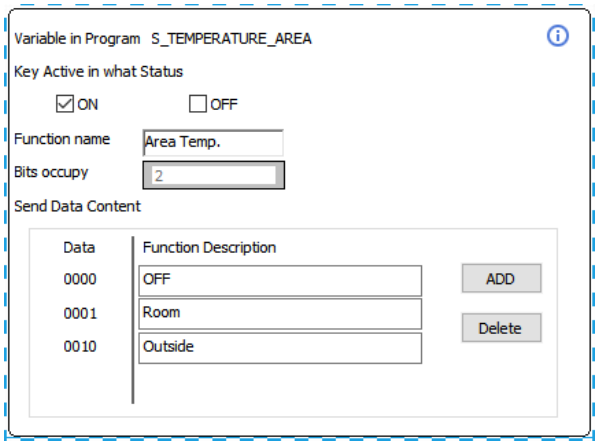


Fig.30 Shift Type Temperature Area Key Configuration

The area temperature function is generally configured to execute key actions in the power-on state. In the sending data content design, various positions can be added using the Add button, with up to 4 areas. Users can annotate the function of each control status bit. When the area temperature function is combined with an LCD display, Level 0 is no display, Level 1~3 is always on in a certain segment.

- Timer

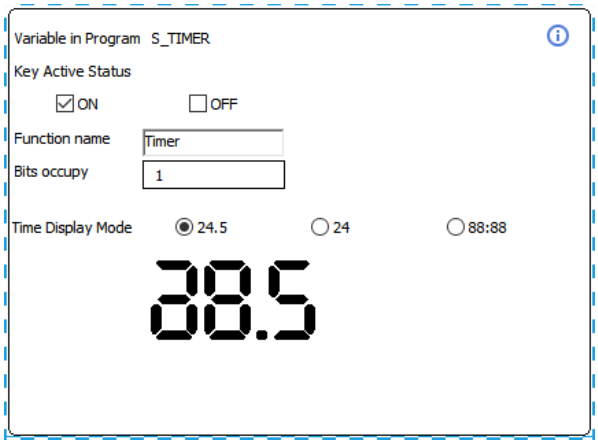


Fig.31 Timer Key Configuration

The timer function is generally configured to execute key actions in both the power-on and power-off states. The timer function is designed so that when the remote controller is in the power-on state, the timer key enables a timed power-off. When the remote controller is in the power-off state, the timer key enables a timed power-on. When the timer function has been enabled, if the on/off key is pressed, the timing mode will automatically exit. The time display mode can be selected to be either 24.5, 24 or 88:88, according to different LCD specifications. The bit length of the timer function is used to store the number of hours or half hours. It is recommended to set this to 6~8 bit length.

- Sweep Check

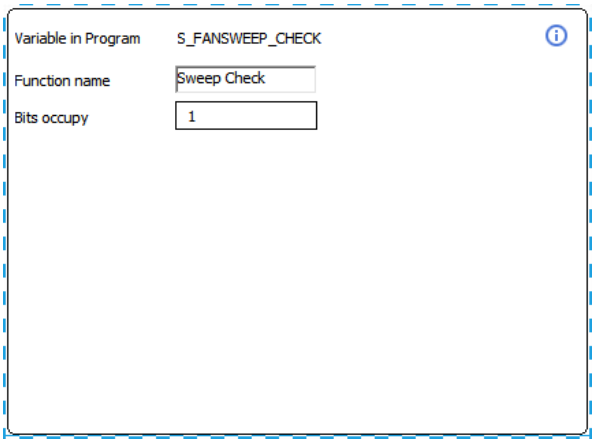


Fig.32 Sweep Data

The sweep check in the code is used to associate the sweep up and down as well as sweep left and right functions and to operate as a confirmation bit, which is a pure data content. If this data function is enabled, the remote controller program will determine whether the sweep up and down or the sweep left and right has been enabled. If so, the sweep check bit will be set to 1. The sweep check can be set to have a 1 to 8 bit length.

- Verify Code, Verify2

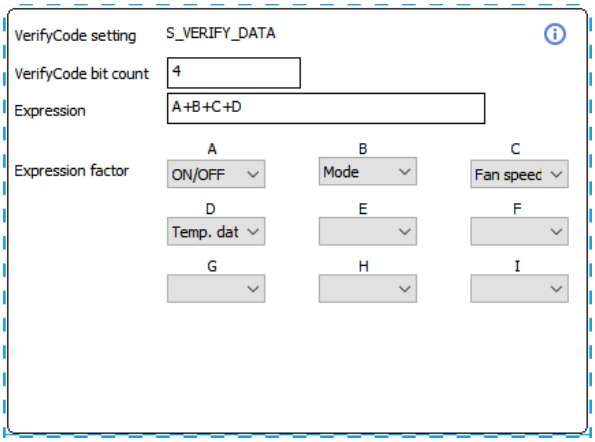


Fig.33 Verify Code Configuration

The verify code in the code is used for verification, which is a pure data content. To configure a check code, users can select the configured sending data as an expression factor, and then add it to the formula in the expression bar to calculate the verify code. Expressions can use addition, subtraction, multiplication and division for this calculation. The verify2 is used for a second check calculation when the second set of codes is enabled. The verify codes can be set to have a 1 to 8 bit length.

- Constant Code 1~10

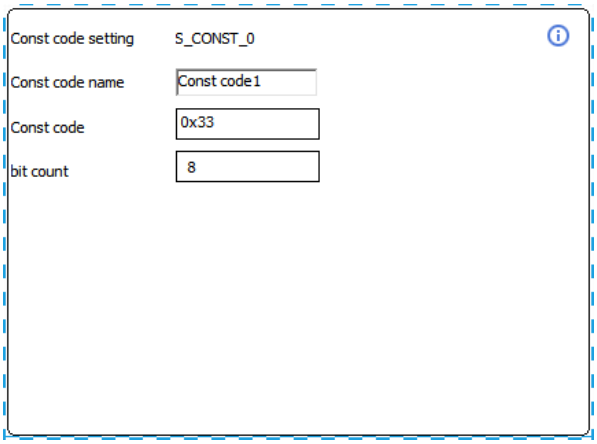


Fig.34 Constant Code Configuration

The constant code is used for verification which has a pure data content. To configure the constant code, it is only necessary to fill in the constant code content and bit count. The constant code can be set to have a length of between 1~8 bits.

- Timer On, Timer Off

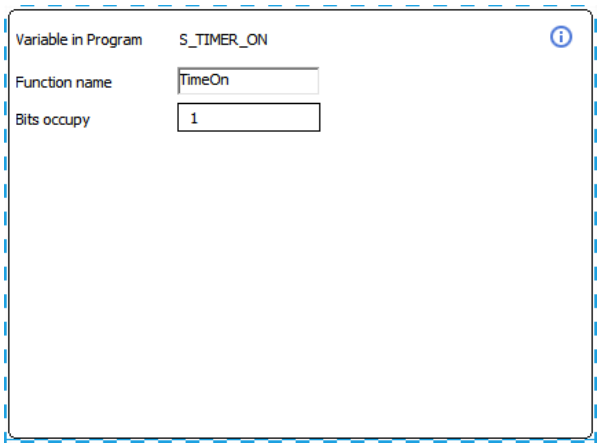


Fig.35 Timer On/Off Associated Data Configuration

The timer on and timer off in the code are used as confirmation bits for the timed power-on and timed power-off operations, which have a pure data content. If these data functions are enabled, the remote controller program will determine whether the timer on and timer off functions have been enabled. If so, the corresponding send bit will be set to 1. The timer on and timer off can be set to have a length of between 1~8 bits.

- Timer On Counter, Timer Off Counter

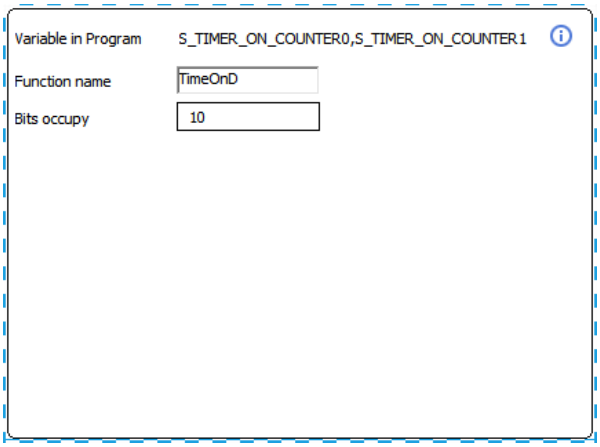


Fig.36 Timer On/Off Counter Configuration

The timer on counter and timer off counter in the code are used to confirm the timer on and off operations which have a pure data content. If these data functions are enabled, the remote controller program will determine whether the timer on and timer off functions are enabled. If so, the remaining time will be written into the corresponding variable. The timer on counter and timer off counter functions correspond to two 8-bit variables which can be set to have a length of between 9~16 bits.

2.1.2.3 Other Configurations

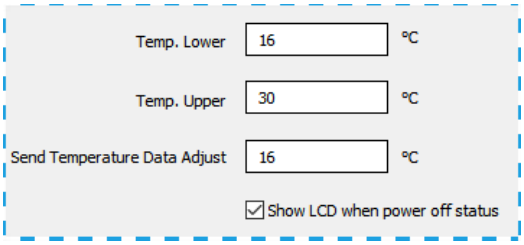


Fig.37 Send Temperature Adjustment Data Configuration

As the temperature adjustment range for most air conditioners is generally between 16°C to 30°C, the F/W in the remote controller can limit the maximum and minimum values in the temperature setting. The temperature deduction base can also be set to reduce the length of the sent temperature. For example, a binary representation of 30 degrees is 11110B, which requires at least 5 bits. After subtracting 16, it becomes 1110B in binary. Therefore a reduction of a single bit is achieved which will reduce the transmission power consumption.

Most LCD remote controllers will maintain their LCD display state when they are turned off. The integrated remote controller scheme in the software can keep the standby current consumption to within 10μA in the LCD display state. If there exists severe current consumption control requirements, users can disable the "show LCD when power off status" function.

2.2.2 LCD Configuration

The LCD display is one of the LCD remote controller components. In the LCD configuration page, the software integrates the common display functions and modes of most LCD remote controllers for LCD air conditioner applications.

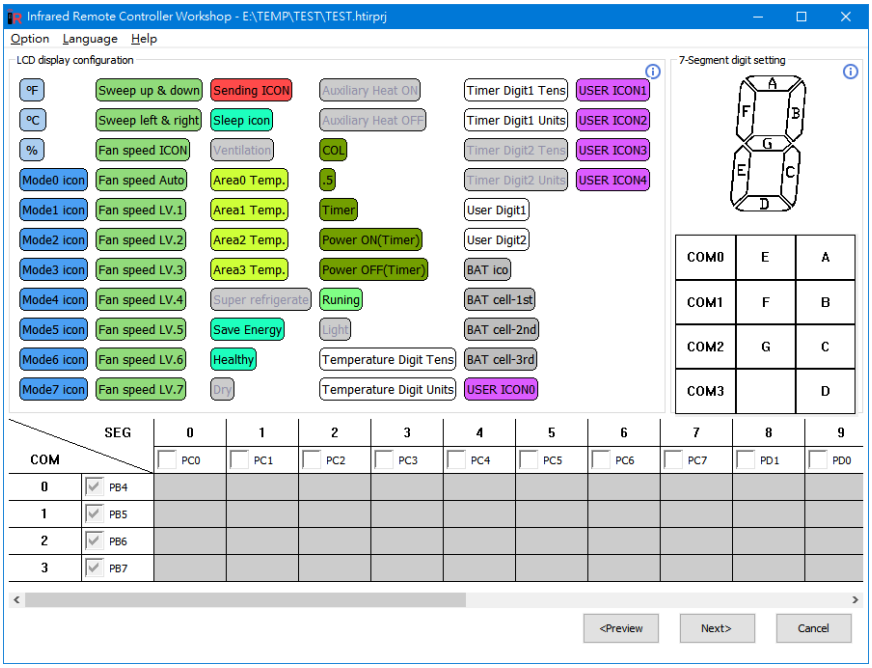


Fig.38 LCD Display Overall Configuration

The top left of the LCD configuration page is an LCD content area to be configured, the right side is a digit adjustment area, and the lower part is the LCD RAM mapping configuration area.

LCD Display Configuration Area

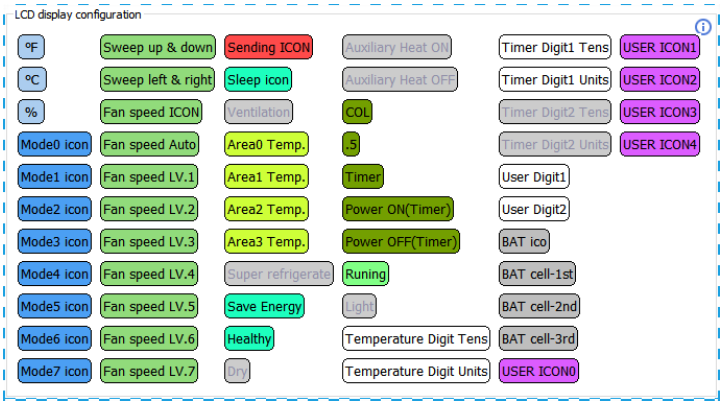


Fig.39 LCD Display Configuration Area

The functions and text that have been configured in the previous step are shown in colours to indicate the available state. If a function is not configured, it will be shown in grey to indicate a restricted state. Drag a display block that has been enabled into the RAM mapping configuration area below to enable the corresponding display function.

Digit Adjustment Area

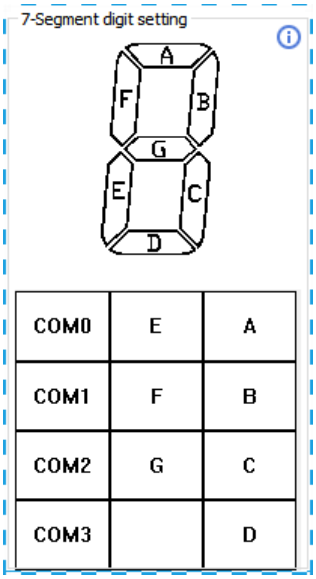


Fig.40 LCD Digit Adjustment Area

The LCD remote controllers generally include a digit display. The 4 COM pins for various LCD display specifications are mostly designed so that the A~G segments of a digit are controlled by two SEGs. These seven segments may have different arrangements due to the use of different LCD display specifications. For different digit display control, users can adjust the segment arrangements in the digit adjustment area to be compatible with different LCD screens. The operation method is shown in the following figure. Click "A" first, and then click "F" which needs to be exchanged, now the positions of "A" and "F" can be exchanged. The other segments can also be exchanged to the corresponding driver positions in this way.

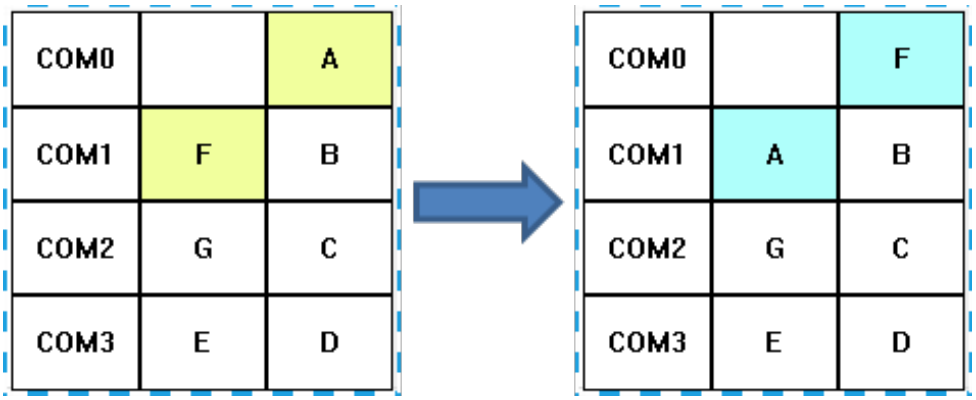
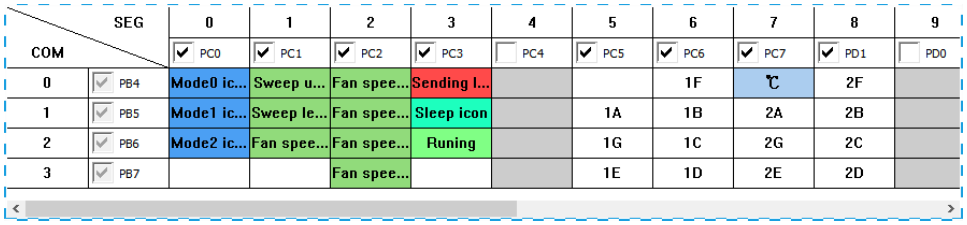


Fig.41 Exchange Method in LCD Digit Adjustment Area

RAM Mapping Area



SEG / COM		0	1	2	3	4	5	6	7	8	9
		PC0	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PD1	PD0
0	PB4	Mode0 ic...	Sweep u...	Fan spee...	Sending I...			1F	℃	2F	
1	PB5	Mode1 ic...	Sweep le...	Fan spee...	Sleep icon		1A	1B	2A	2B	
2	PB6	Mode2 ic...	Fan spee...	Fan spee...	Runing		1G	1C	2G	2C	
3	PB7			Fan spee...			1E	1D	2E	2D	

Fig.42 LCD RAM Relational Table

The left column of the RAM mapping area is the COM pin configuration. The COM pins are arranged according to the HT67F2432 COM0~COM3 pins. This function is compulsory enabled. The top row is the SEG pin configuration. The SEG pins follow the configuration of the HT67F2432 SEG0~SEG19. The SEG0~SEG19 pins are mapped to addresses 00H~13H respectively in RAM Bank4. The SEG13 pin (PA4) is pin-shared with the REM pin, so the software forces the LCD driver function of SEG13 to be disabled.

The digit display has seven segments. Therefore, when dragging a digit into the RAM area, the software will automatically allocate two continuous SEGs and take up seven bits of the RAM Bank4 space. If the digits require different non-continuous SEG pins for driving, users can also drag any set of SEGs to another position.

For part of LCD display, some ten digit numbers do not need to display the F segment, users can right-click it to remove the F segment after a digit is configured to the RAM mapping area.

The corresponding I/O pin name is shown below each SEG. When the segment to be displayed is dragged into the RAM configuration area, the software will automatically enable the SEG pin if it has not been enabled. If an I/O is not used for the LCD driver, the LCD driver pin function should be disabled. Otherwise, this I/O cannot be configured as a key in the next operation.

2.2.3 Step Matrix Keys and Driver

When the LCD remote controller protocol and LCD functions have been configured, the software will jump to the next configuration page for the matrix keys. As shown in the following figure, area A is an I/O area for selection, area B is a supported MCU package prompt area for the current configuration scheme, area C is an IR LED driver mode display area, area D is a configured key area, area E is a key area of the generated keyboard and area F is a function key area to be configured.

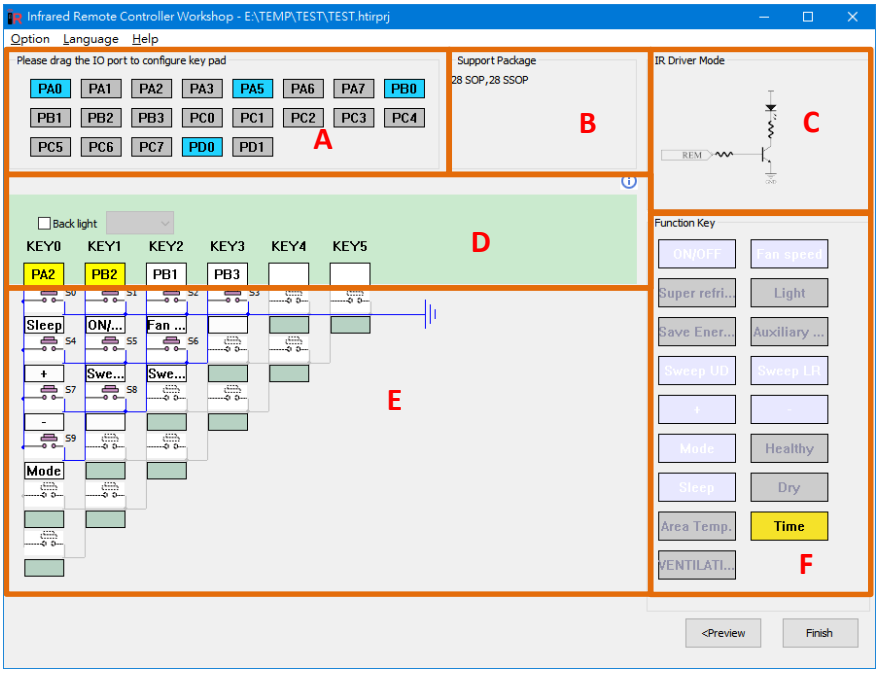


Fig.43 Key Configuration Area

Area A, at the top left of the configuration page, shows the available I/O pins. Here the blue pin icon is common to the 24SOP/24SSOP and 28SOP/28SSOP device packages while the green pin icon is unique to the 28SOP/28SSOP packages. The I/O configuration can be implemented by dragging an I/O icon into the key position below area D. Area B, at the top of the key page, indicates the available packages according to the current key configuration. If the PC0, PC1, PC2, and PC3 pins are configured as LCD driver pins in the LCD configuration page or are dragged into area D to be configured as key pins, the 24SOP/24SSOP packages in area B will be removed.

When the I/O pins are configured to the key configuration table, a step matrix keyboard is generated automatically. The maximum number of keys that can be generated is calculated using the formula: K = 1+2+...+N, where N is the number of the configured I/O pins. The HT67F2432 LCD remote controller can configure up to 21 keys using 6 I/O pins.

Similarly, when clicking on any configurable key table, the two I/Os connected with this key will be highlighted in yellow.

When clicking on the key table on the first row, only one I/O will be highlighted in yellow. This is because the keys corresponding to this row consists of I/Os and VSS.

The final key function configuration can be implemented by dragging the corresponding function block from area F to the key in area E on the left.

The HT67F2432 only supports the REM mode to drive the output, therefore area C is only used to show the drive circuit, here the drive mode cannot be changed.

3. Workshop Hardware

The supporting hardware for the Infrared Remote Controller Workshop consists of three parts: two development boards, ESK-IRRC-T00 and ESK-IRRC-T01 and a decoder board, ESK-IRRC-R00.

3.1 ESK-IRRC-T00

The ESK-IRRC-T00 is a general remote controller development board using the HT68F2420 in a 20SSOP package as the master controller. It uses 16 I/O pins to form a 136-key step matrix keyboard, which can emulate any I/O combination of keys. The REMDRV pin is supplied to directly drive a 5mm IR LED. The power supply uses CR2032 batteries. The circuit is shown below.

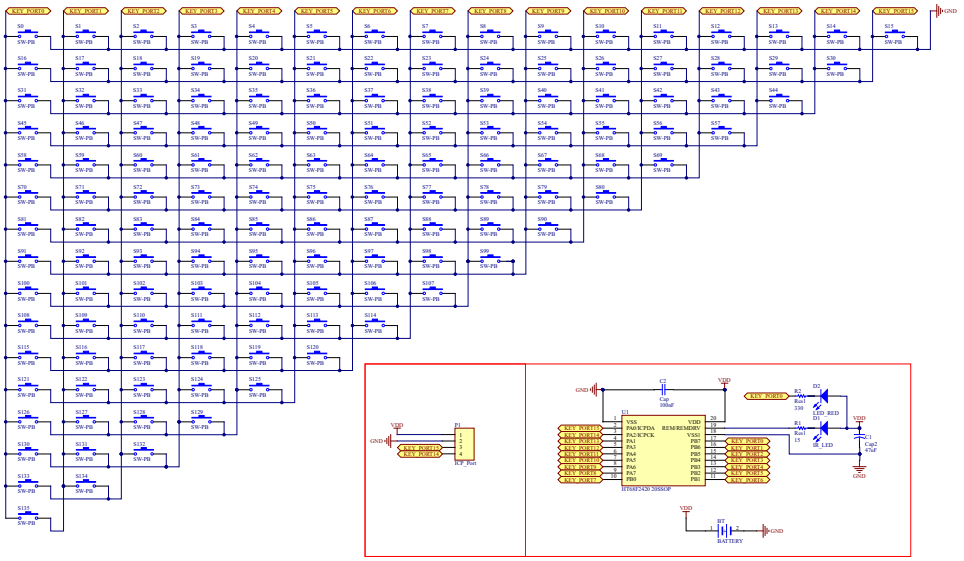


Fig.44 ESK-IRRC-T00 Schematic Diagram

3.2 ESK-IRRC-T01

The ESK-IRRC-T01 is an LCD remote controller development board using the HT67F2432 in a 28SSOP package as the master controller. It uses 5 I/O pins to form a 15-key step matrix keyboard and 20 I/O pins to configure a 4×16 LCD driver. The REM pin is used to drive a MOS transistor which in turn drives a 5mm IR LED. The power supply uses CR2032 batteries. The circuit is shown below.

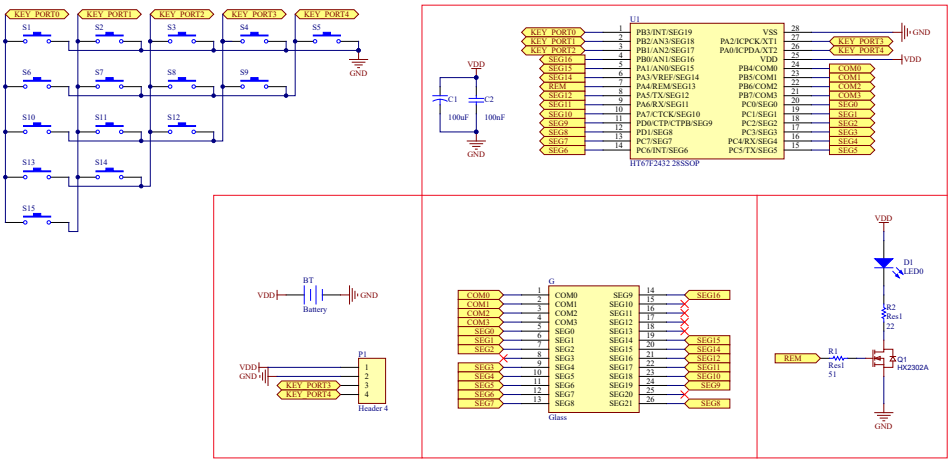


Fig.45 ESK-IRRC-T01 Schematic Diagram

3.3 ESK-IRRC-R00

The ESK-IRRC-R00 is a decoder board using the HT66FB550 as the master controller, which captures infrared carrier signals at a rate of 16MHz. The captured signal is uploaded to a computer via a Micro USB interface on the board and the decoded content is displayed on the workshop software. The decoder board is an HID device which does not require any additional drivers to be installed when being connected to a computer.

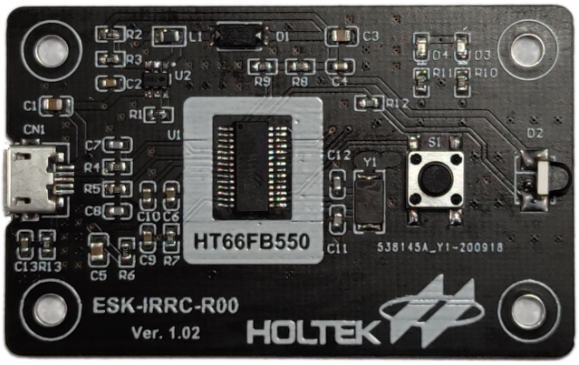


Fig.46 ESK-IRRC-R00 Decoder Board Hardware Diagram

After the decoder board is connected to a computer, both the red and green LEDs will turn on. When the computer has been identified the USB device successfully, the decoder board green LED will flash every 2 seconds.

The right side of the decoder board has the receiving induction probe which is used to sense infrared remote controller signals.

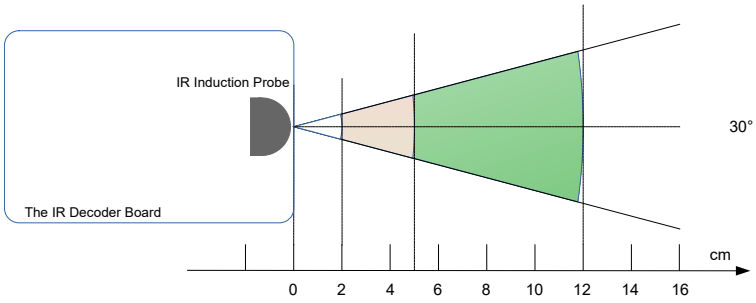


Fig.47 Receiving Top View

To ensure that the decoder board can operate in the best way possible, it is recommended that the angle from the remote controller transmitter to the decoder board induction probe should be no more than ±10 degrees up and down and no more than ±15 degrees left and right. In general, the optimal receiving distance should be between 5cm and 12cm (about 1 palm wide). For some remote controllers with lower power transmitters, the sending code distance can be reduced appropriately. A distance between 2cm and 4cm has the strongest code identification ability but this distance will make the identified carrier duty larger than it really is.

4. Decoding and Learning Code Development

4.1 Code Verification

After a remote controller with standard protocol or a user defined protocol development has completed, the generated project can be compiled and programmed to the remote controller development board. The remote controller development board program can then be verified by sending the code to the decoder board using the development board.

The operations are as follows:

1. Connect the decoder board to a computer.
2. Select new project and "code/waveform mode" then jump to the software view as shown in Fig.48.
3. Press the "start decode" button in the software. Now the decoder board will enter a receiving state and the red LED will turn on. The remote controller to be tested should now be aimed at the decoder board induction probe within 6 seconds and press a remote controller button to emit an infrared waveform.
4. When the decoder board green LED turns on, this indicates that the reception has completed and that the sending data will be uploaded to the computer. When the upload is successful, the decoder board green LED will flash 4 times and the sending parameters will be displayed after the decoding has completed.
5. If the sending code is not received or the data is abnormal, the decoder board red LED will flash 4 times to indicate that the upload has failed.

The general remote controller decode receiving results are shown below:

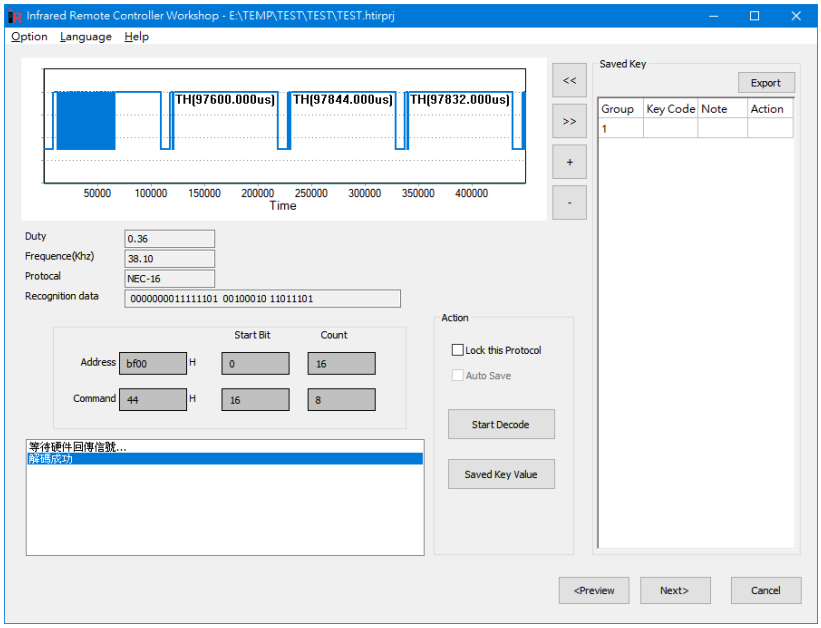


Fig.48 General Remote Controller Receiving Verification

In Fig.48, the upper part of the display shows the remote controller code waveform. The waveform is similar to an IR receiver device 1838 output mode. The MARK with carrier is displayed with a low level and the SPACE without carrier is displayed with a high level. The right side of the waveform has buttons to zoom in or out and move left or right. Point to the waveform and hold down the left mouse button to draw a rectangle on the lower right and the graph will zoom in. While

holding down the left mouse button to draw a rectangle on the upper right, the graph can be zoomed out. The graph can be dragged left or right for viewing by holding down the right mouse button.

The contents below the waveform diagram show the specific parameters of the sending code, such as the carrier duty, frequency, the identified protocol type and each bit recorded by the waveform. Below this string of bits are the identified address and command as well as their start bit and total bit count. Only when a standard protocol has been identified will the software automatically identify the start bit and the total bit count of the address and command according to the protocol type. If the sending code does not conform to a standard protocol, the software will determine that it is a user defined protocol, which requires users to manually enter the start bit and total bit count of the address and command.

The LCD remote controller decoding effect is shown below:

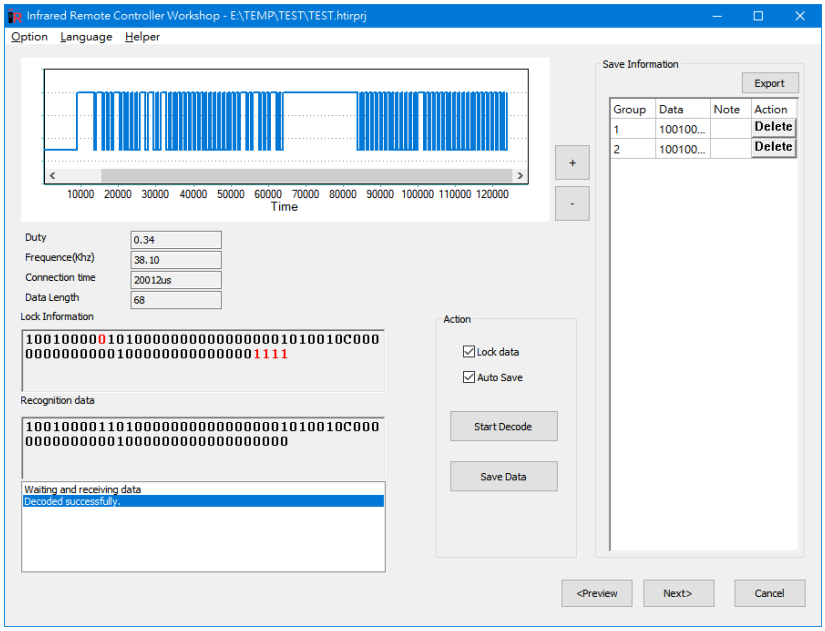


Fig.49 LCD Remote Controller Receiving Verification

In Fig.49, the upper part of the display shows the remote controller code waveform. The waveform is similar to an IR receiver device 1838 output mode. The MARK with carrier is displayed with a low level and the SPACE without carrier is displayed with a high level. The right side of the waveform has buttons to zoom in or out and move left or right. Point to the waveform and hold down the left mouse button to draw a rectangle on the lower right and the graph will zoom in. While holding down the left mouse button to draw a rectangle on the upper right, the graph can be zoomed out. The graph can be dragged left or right for viewing by holding down the right mouse button.

The contents below the waveform diagram are the specific parameters of the sending code, such as the carrier duty, frequency, connect time, total length of data code.

The default recognition mode in the recognition data area is PDM modulation. A bit is identified as 1 when it has longer SPACE time or as 0 if its SPACE time is shorter. When Lock Data is checked, this string of data will be copied to the lock area and each new identified code will be compared with the lock area. When any bit is different from the lock area, the binary data corresponding to the bit in the lock area will be displayed in red. For example, as shown in the figure above, the new identified code is 1 degree greater than the lock area, showing two changes. Therefore, the temperature bit can be determined and the temperature data can be added to the check code calculation.

In the content of the identification data area, C means a connect code is inserted here, L means an inserted interval between two codes and E means the end of the sending code.

4.2 Code Development

If there is an existing remote controller and the original project or source code cannot be found, but a scheme with the same protocol as this remote controller needs to be developed, in this case the code development mode can be used. Use a decoder board to read the remote controller protocol, carrier, address, command and other parameters to develop remote controller products with consistent protocol and functions.

4.2.1 General Remote Controller Code Development

After the above code verification steps are executed and the infrared remote controller code is analysed, click the "Saved Key Value" button to save the recognised command for key use in new schemes.

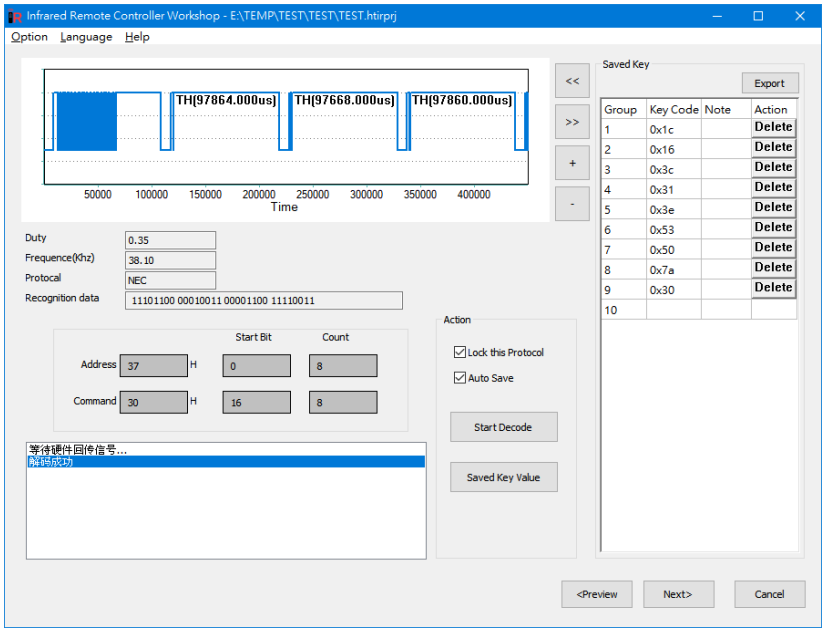


Fig.50 Receiving Code Development

For non-standard protocol code, as the bit count of the address or command cannot be predetermined by the software, users should analyse multiple groups of code to determine their start bit and total bit count and manually enter them into the configuration bar. The software can identify the correct code according to the address and command code.

If multiple key command codes are required to be saved, "Lock this Protocol" and "Auto Save" can be checked after the protocol is correctly recognised, which can simplify the multi-key learning operations.

After the command code of several keys has been successfully saved, click Next to jump to the corresponding standard protocol or user defined protocol and then modify some of the parameters.

1. If a standard protocol is identified, then Next will jump to the standard protocol parameter configuration page, here only the address and duty can be changed. The software will pre-write the address code of the learned remote controller.

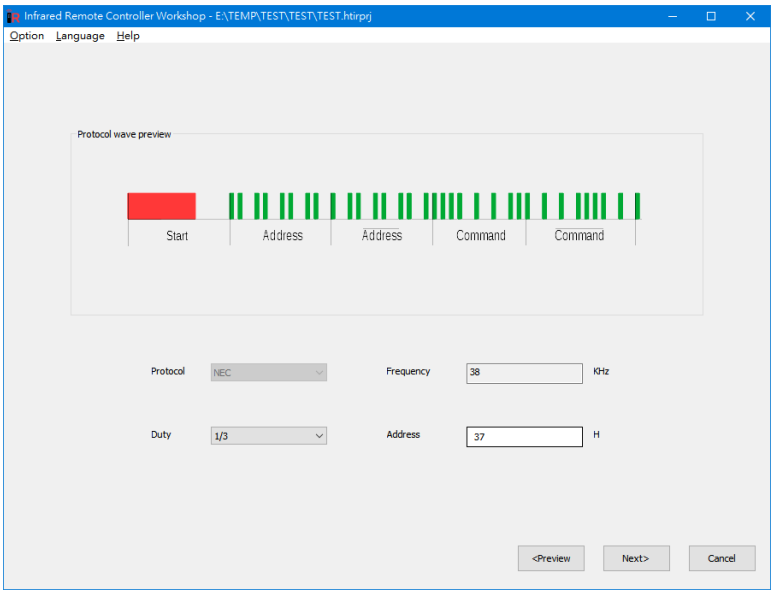


Fig.51 Code Standard Protocol Jumped Page

2. If a non-standard protocol is identified, the software will import the relevant code type parameters by automatically recording it into the user defined protocol design parameters. Many remote controller's sending code is influenced by software instructions, so the actual measured parameters will be slightly different from the design parameters. Therefore, in the development of converting a non-standard protocol into user defined parameters, it is recommended to appropriately adjust the software imported parameters.

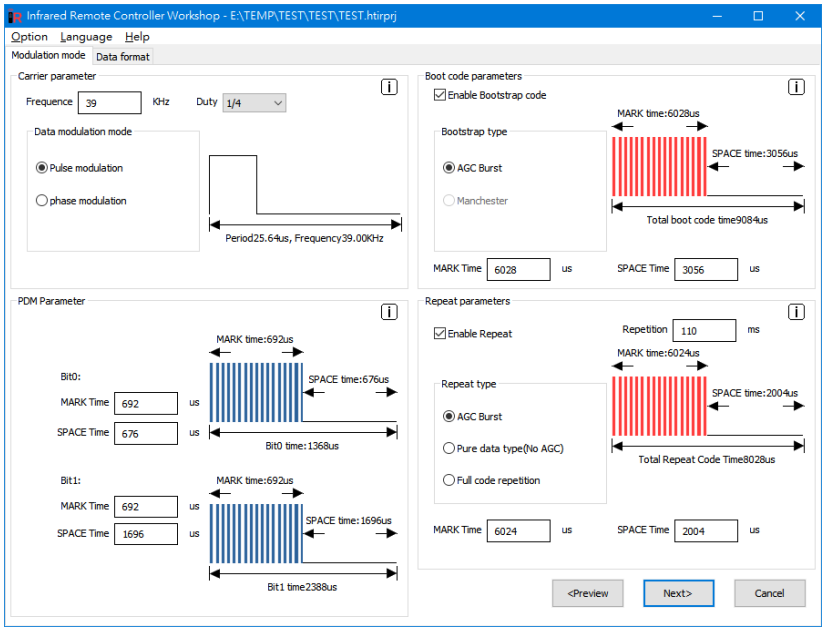


Fig.52 Code Non-standard Protocol Jumped Page

4.2.2 LCD Remote Controller Code Development

The identification operation follows the code verification steps. After the infrared remote controller code parameters and data are analysed, click the "Saved Data" button to save the recognised data.

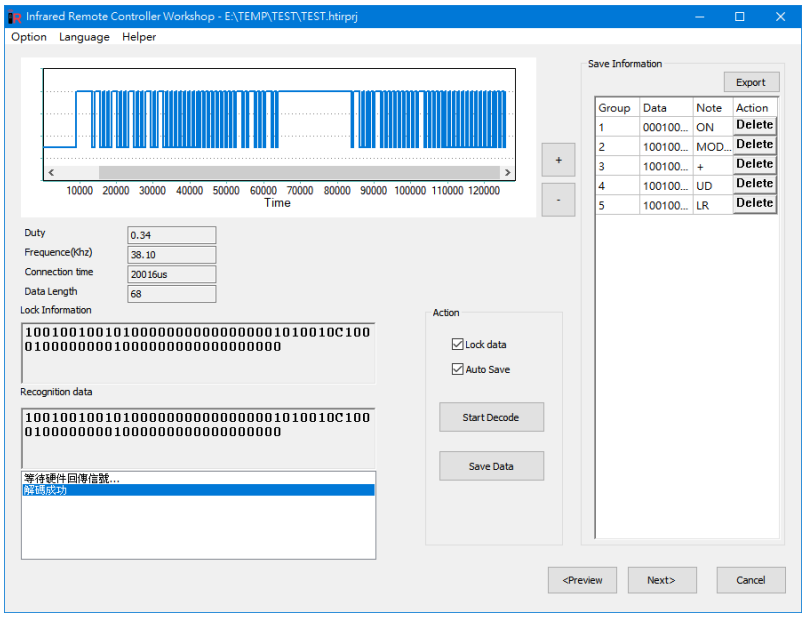


Fig.53 LCD Remote Controller Decoding

The upper right corner of the software has an Export button which can save the code parameters and the saved data code of the LCD remote controller in the form of a CSV table.

Create time:		"2021-04-27 16:13:03		
Decoding Frequency:		'38.10		
Duty:		'0.34		
Data Length(not contain Header):		'68		
Carrier:		'		
	MARK	'9072		
	SPACE	'4476		
Bit0		'		
	MARK	'612		
	SPACE	'600		
Bit1		'		
	MARK	'612		
	SPACE	'1568		
Connect code:		'1		
	Connect Time:	'20ms		
Identify Data(The first received cc		'		
	Group1:	'000100001001000000000000000001010010C0000000000001000000000000001011		ON
	Group2:	'100100001001000000000000000001010010C0000000000001000000000000000111		MODE1
	Group3:	'100100000101000000000000000001010010C0000000000001000000000000001111		+
	Group4:	'100100100101000000000000000001010010C1000000000001000000000000001111		UD
	Group5:	'100100100101000000000000000001010010C1000100000001000000000000000000		LR

Fig.54 Export Parameters and Contents for LCD Remote Controller Decoding

The LCD remote controller sending code data content is complex and difficult to be predetermined, so the learning code software does not analyse the data content. It is up to users to compare the differences in the code of each key function. After the completion of the learning code, click Next and the software will display the following prompt. The LCD remote controller learning operation only imports decoding code parameters and does not produce any functional configuration.

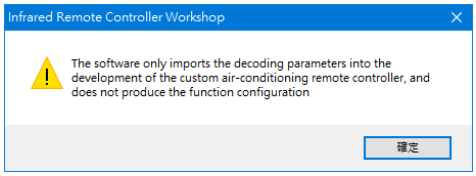


Fig.55 LCD Remote Controller Decoding Prompt

5. Description of Other Functions

5.1 F/W Power Control

The Infrared Remote Controller Workshop has been especially designed for remote controllers. The power consumption for the remote controllers has been optimised as much as possible.

5.1.1 Dynamic Power Control

In the market, some remote controllers have the IR LED completely on during the SPACE time, which is not good for battery life. The workshop generated remote controller program can implement a situation where the IR LED is only driven by the MARK duty time and is completely off during the SPACE control time, greatly extending the battery life.

General remote controllers normally need to send repeat codes due to their protocol. They will continue to send codes after their buttons are pressed. Most remote controllers do not set a time threshold. If any remote controller buttons are accidentally pressed by nearby objects, this will cause the remote controller to send repeat codes until the battery energy is exhausted. However, this workshop's firmware can determine the number of repeat code sending times. When the remote control enters a press button state, the repeat codes will only be repeatedly sent 255 times. When this preset threshold is reached, the program will turn off the IR LED emission until the button is released to reduce power consumption.

5.1.2 Static Power Control

Using the F/W generated remote controller program, the general remote controller is woken up from the HALT mode by a WDT overflow every 0.128s to scan the keys. While the LCD remote controller is woken up by an interrupt every 64ms to check the timer, it can scan the keys every 0.128 seconds and execute an LCD flash function, backlight timing or battery voltage detection (open in secondary development only) every 0.512 seconds. General remote controllers can scan up to 136 keys, and the standby power consumption can be controlled to within 7μA. For LCD remote controllers, the power consumption can be controlled to within 10μA when the LCD is always on and to within 5μA when the LCD is off.

5.2 Decode Recognition

The decoder board can identify code lengths of up to 148 combinations of MARK and SPACE which basically covers the codes of all general remote controllers and most LCD remote controllers.

As there are many remote controller protocols and code sending protocols in the market, some special protocols that do not conform to general coding rules may be difficult to identify in the workshop, therefore it is not guaranteed that every protocol can be recognised and restored. In the future, updated versions of the software will continue to enhance these learning functions and attempt to increase the range of available protocols.

6. Appendix & FAQ

6.1 LCD Remote Controller Code Control and Data Table

Function	Independent Key	Data	Key Active in Power-On/Off Status	Associated Keys	Function Special Description
Power on/off	√	√	√	ON/OFF	Clear all timing states
Mode	√	√	×	Mode	Switching modes will automatically clear super refrigeration, sleep, and energy saving states
Fan speed	√	√	×	Fan speed	The default Level 0 is the auto mode
Temperature data	×	√	×	[+], [-]	
Sweep up and down	√	√	×	Sweep UD	
Sweep left and right	√	√	×	Sweep LR	
Sweep check	×	√	×	Sweep UD, Sweep LR	
Super refrigerate	√	√	×	Super refrigerate	
Save energy	√	√	×	Save energy	
Healthy	√	√	×	Healthy	
Sleep	√	√	×	Sleep	
Ventilation	√	√	×	Ventilation	
Dry	√	√	×	Dry	
Auxiliary heat	√	√	×	Auxiliary heat	
Light	√	√	×	Light	
Area temperature	√	√	×	Area Temp.	
Verify code	×	√	×	*	
Timer	√	√	√	Timer, [+], [-]	
Timer on	×	√	×	Timer	
Timer off	×	√	×	Timer	
Timer on counter	×	√	×	Timer	
Timer off counter	×	√	×	Timer	
Constant code 0~10	×	√	×	N/A	

* The verify code can use the code value of any key according to the calculation formula.

6.2 Special Modulation Code Description and Applications

In infrared remote controller modulation, some of the infrared code conforms to a specific protocol and can be decoded using the corresponding protocol specified decoder program. However, there are still some identification problems when the decoder board of the platform cannot predetermine the protocol which is used for decoding recognition. Specific cases are as follows.

Modulation Mode	Special Case	Decoding Exception Description	Solution
PDM	All bits are full 1 or full 0.	Because only one bit-length pattern is recognised after decoding, the decoding function configures it as 1.	The address and key should avoid using the code with full 1 or full 0.
PDM NEC-16	The address is configured as 55AAH or 00FFH.	Because the higher 8 bits and the lower 8 bits of 55AAH or 00FFH are the complement of each other and conform to the address encoding mode of the NEC protocol, the decoding identifies it as an NEC code in priority.	When using the NEC-16 protocol, the address code should avoid situations where the higher 8 bits and lower 8 bits are the complement of each other.
Manchester	All bits are full 1 or full 0.	Because only one bit-length pattern is recognised after decoding, the decoding function configures it as 1.	The address and key should avoid using the code with full 1 or full 0; Add a toggle.
Manchester	The start bit is MARK sent last and the end bit is non-MARK sent first. When sending code, all bits are full 10 or full 01.	Because only one bit-length pattern is recognised after decoding, the decoding function configures it as 1.	Add a toggle; The address and key should avoid being configured to 55H or AAH.
Manchester	The header of the user defined protocol which is modulated using the Manchester mode is not effectively recognised as a header when decoded.	Because the Manchester type header is the same as the start, toggle and data bits, it cannot distinguish them from each other during recognition.	This is one of the user defined protocol features and does not affect the decoding of programs dedicated to this protocol.
LCD Remote Controller	All functions are off and the codes are full 0.	This causes the decoder board to recognise only one bit length when all functions are configured to be off and no fixed codes are included or fixed codes are full 0.	The protocol should configure multiple groups of non-continuous fixed codes with non-zero.

6.3 MARK and SPACE Design Error Description

For the decoding of MARK and SPACE by the receiver device, there is an error time of up to period×(1-duty) in the remote controller sending code. This error occurs due to the fact that different protocols have different understandings and designs for the MARK time parameters.

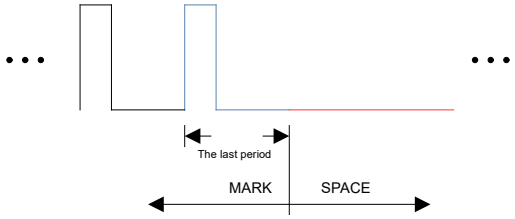


Fig.56 Last Period Processing

For example, when the protocol is designed that the carrier frequency is 38kHz and the duty is 1/3, if the MARK time is set within the range of 560μs~579μs, only the beginning and end of the output pulse have 22 IR pulses displayed on the oscilloscope or logic analyser. In such a case, the same waveform is output with a different MARK definition time. Therefore, the measured value is usually less than the actual MARK waveform time.

The identification time of the MARK last pulse is recorded as a complete carrier time using the decoder board of the remote controller workshop, as shown in Fig.56. Therefore, the identified MARK waveform parameters are larger than the designed ones. It is recommended to make appropriate adjustments to the imported MARK parameters after the codes have been learned.