Infrared Remote Controller Workshop
User Guide

Revision: V1.00 Date: January 19, 2024

www.holtek.com

Table of Contents

1. Introduction

The Holtek Infrared Remote Controller Workshop is a design software 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.

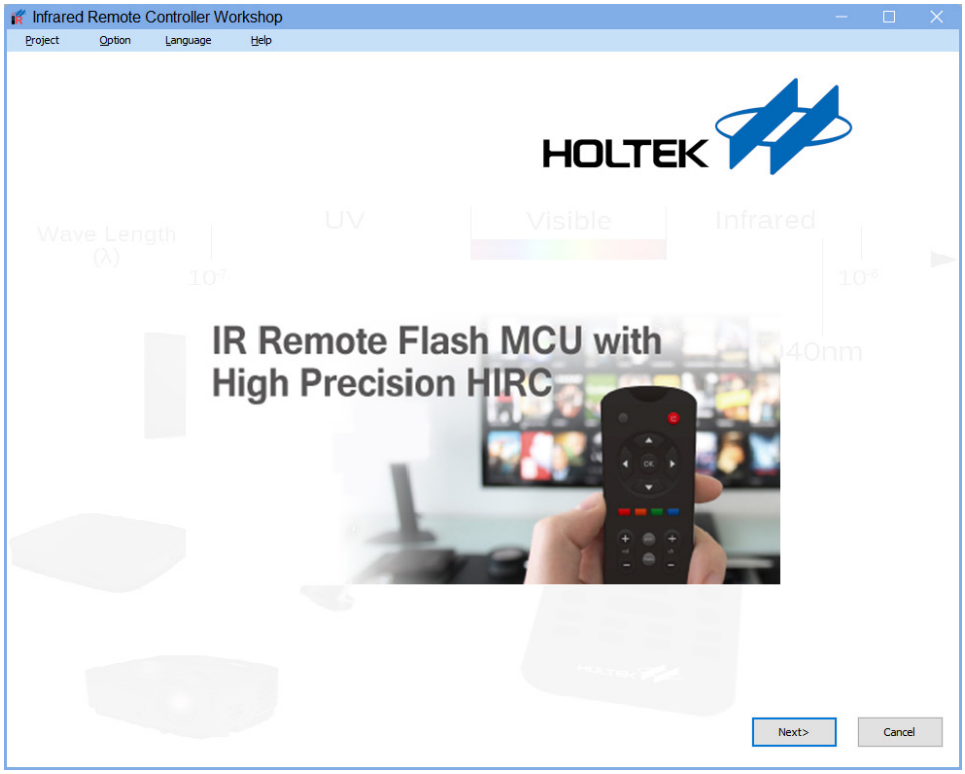
   Code modulation: use different code protocols to modulate signals to avoid interference between different receiver devices.

   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



Boot Screen

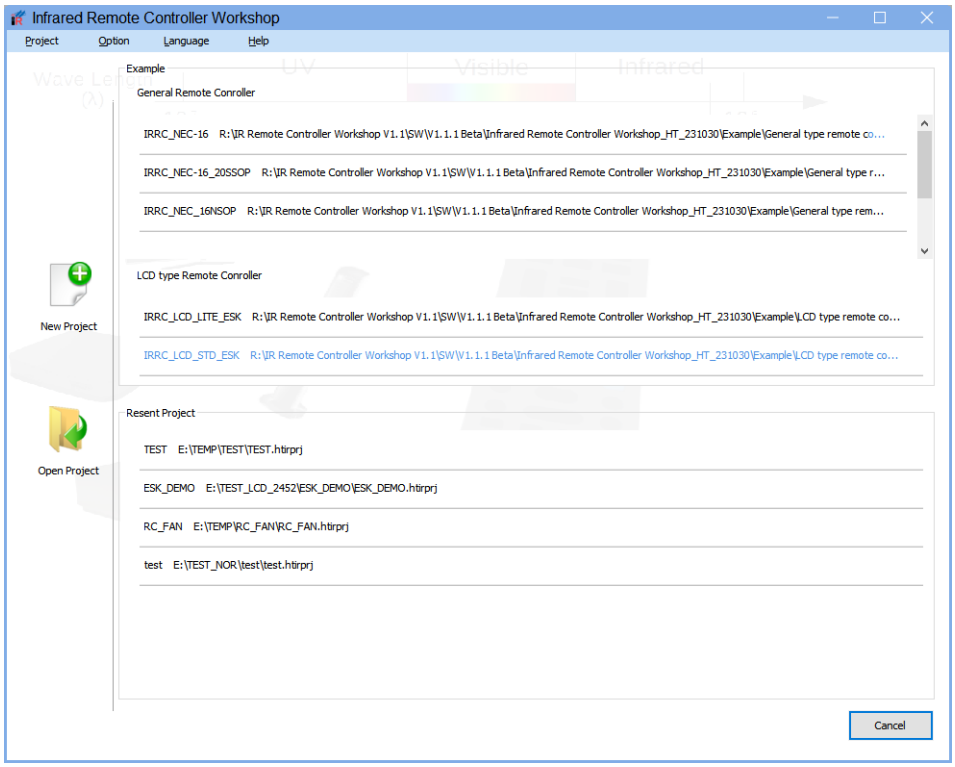
After the software has started as shown above, the menu bar at the top left will show the following menus from left to right: Project, Option, Language and Help.

The Project menu provides New Project and Open Project options.

The Option menu provides a code mode widget 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 guide, development board schematic design, infrared remote controller knowledge base and workshop version information.



Project Management

In the Project Management screen, users can create a new project or open an old project. Several sample solutions are also provided on the software screen for users to refer to.

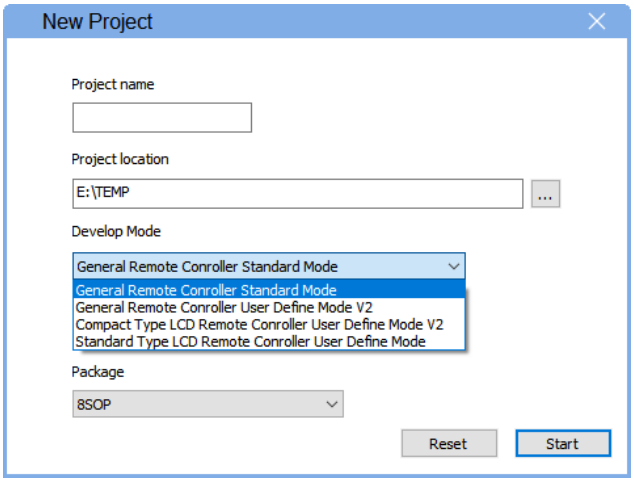
The remote controller development process in the workshop software consists of the following main configuration steps:

a) Protocol definition

b) LCD display content – for LCD remote controllers only

c) Key and driver configurations

d) Generate a project which can be used for secondary development, go to the IDE3000 and then compile it to generate the programming files

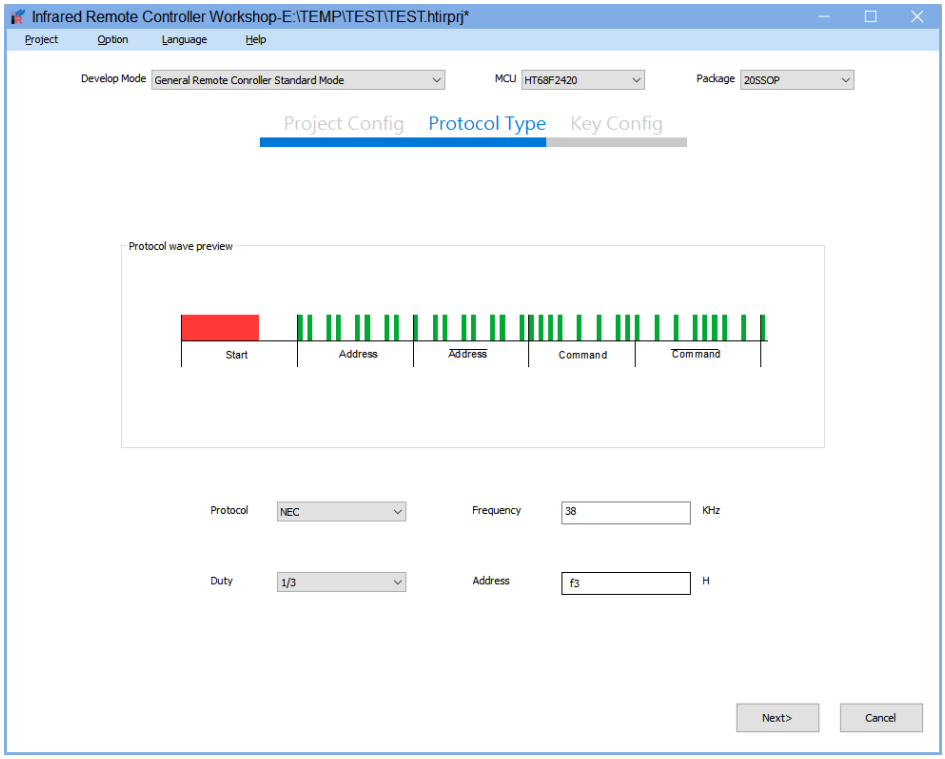


Project Creation

2.1 General Remote Controller Development

The general remote controller development supports two development modes: standard protocol mode and user defined protocol mode.

2.1.1 Standard Protocol Mode Development



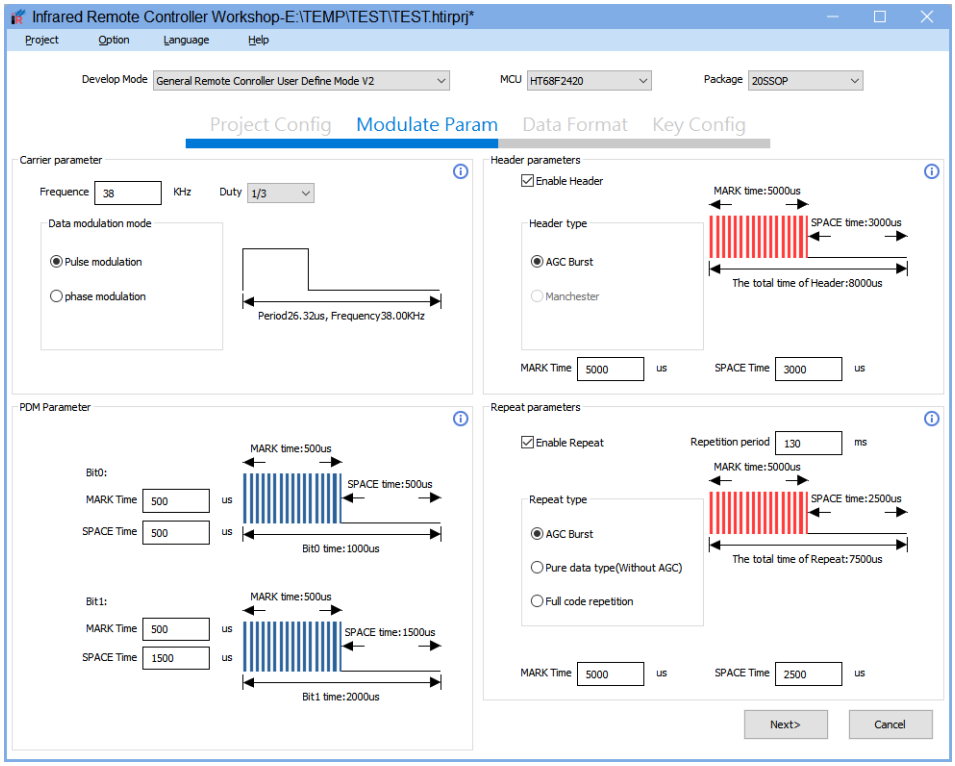
Standard Protocol Mode Configuration

As shown above, the general remote controller standard protocol development is divided into two steps. In the protocol selection step, NEC, NEC-16, Philips RC-5, Philips RC-6, and other popular standard protocols in the market can be selected. 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 general remote controller user defined protocol mode development is divided into three steps.

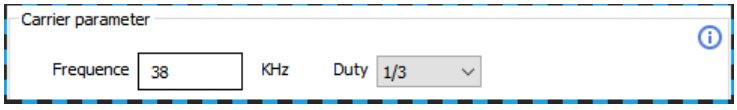
The first step is to set the modulation parameters. In the modulation mode tab, the carrier parameters, Bit1 and Bit0 modulation parameters, boot code parameters and repeat code parameters can be defined by users.



User Defined Protocol Mode Configuration

The specific configuration operations are as follows:

Carrier

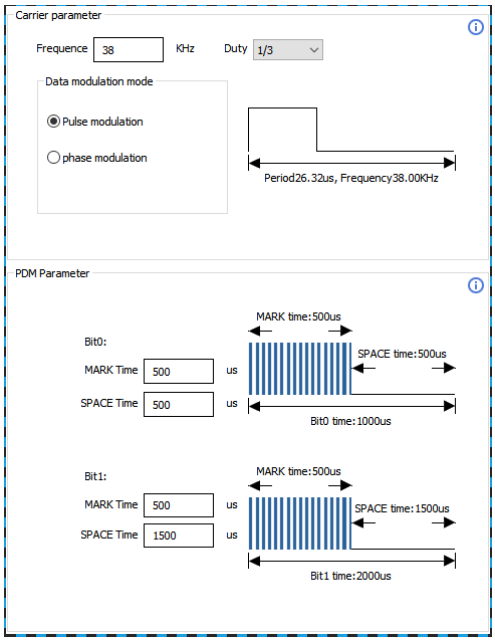


Carrier Parameters

As shown above, 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.

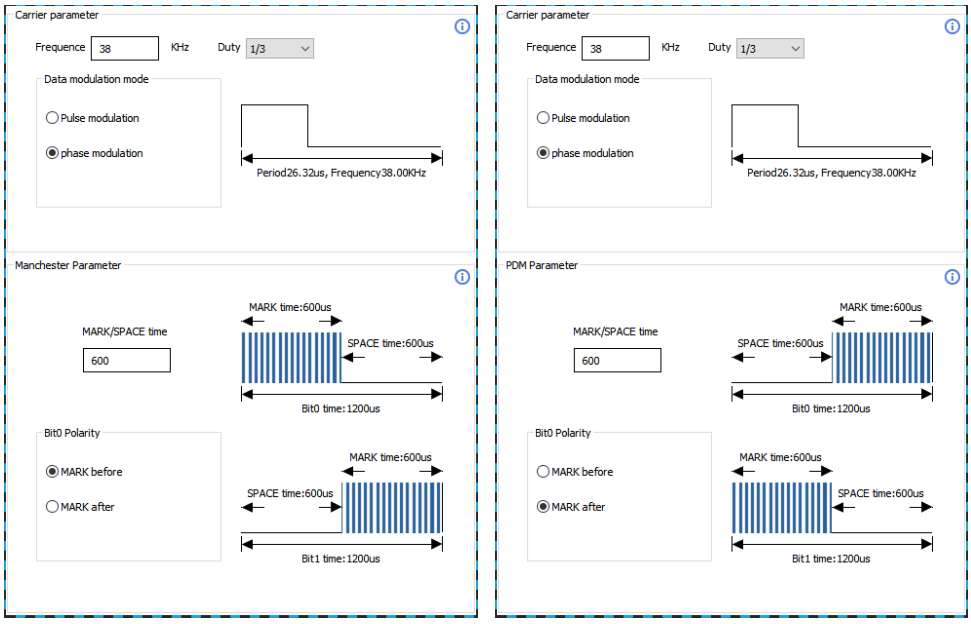
Data Modulation Mode

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



Pulse Duration Modulation (PDM) Parameters

The figure above 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.



Phase Modulation (Manchester) Parameters

The figure above 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,

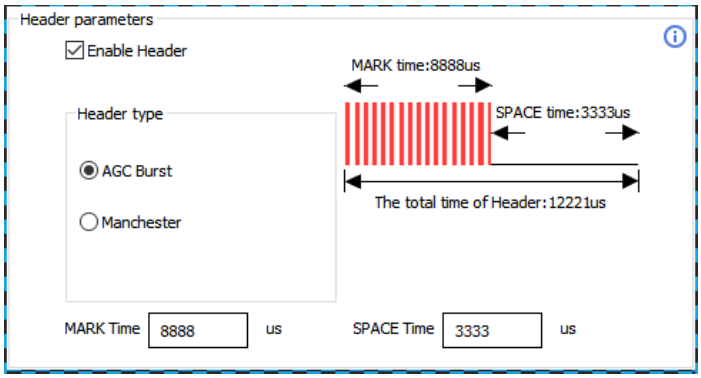
only one time parameter needs to be set. Then use polarity definition to select whether the MARK time of Bit0 is in front or behind. The MARK, SPACE and total time parameters are marked graphically on the right side by the software.

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

Boot Code/Header Parameters

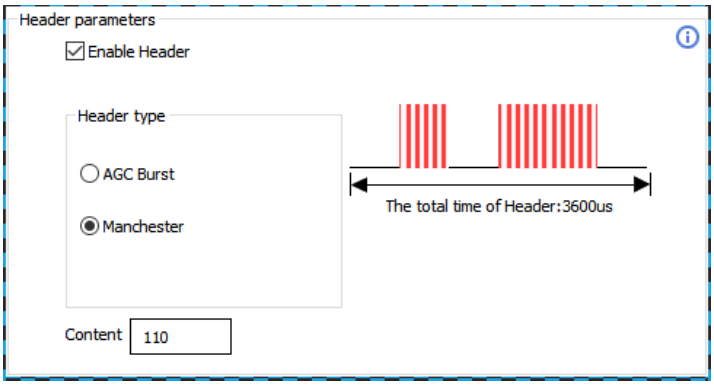
The Boot Code, also known as a Header, is used by the remote controller to send an initial calibration signal before sending the control signals to a receiver.

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.



AGC Burst Type Boot Code Parameters

As shown above, the AGC Burst type boot code consists of a period of MARK and SPACE that is longer than the universal 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.



Manchester Type Boot Code Parameters

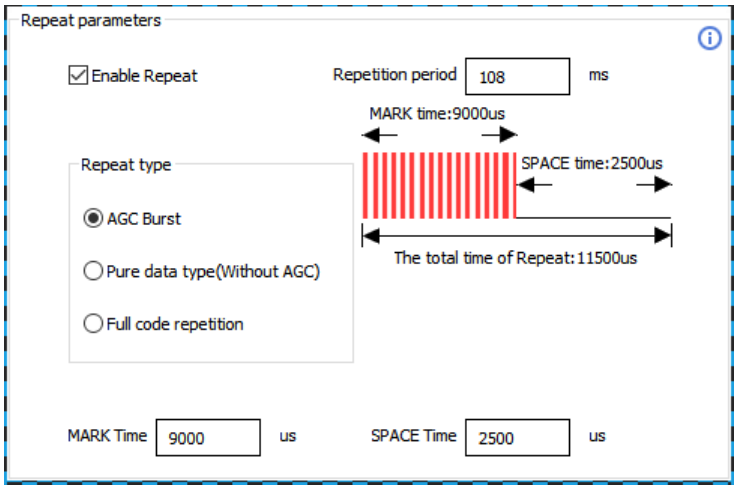
As shown above, when data is modulated using the Manchester mode, a digital boot code similar to the Philips RC-5 protocol can be used. The boot code in the software can be configured with up to 3 bits. The positions of MARK and SPACE as well as total time parameter are marked graphically by the software.

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 below, 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.



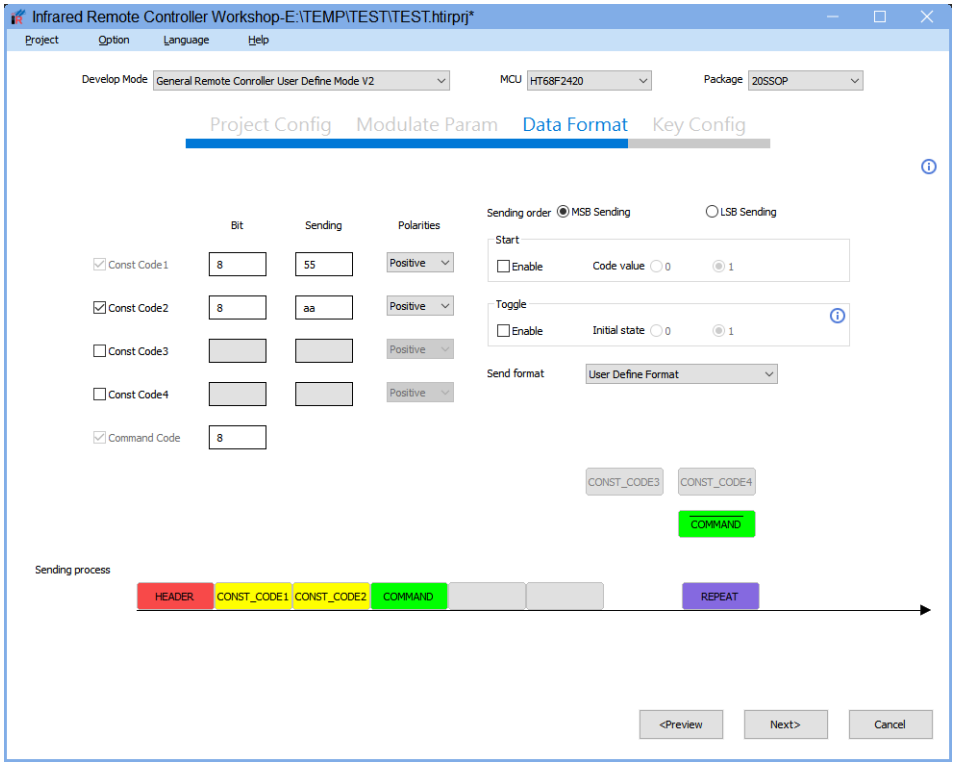
Repeat Code Parameters

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 below, in the second tab, the data format parameters can be set, such as constant code, command code, the sending order and whether to insert Start and Toggle bits or not.



Data Format Parameters

Constant code: 1~4 configurable groups, forced to be enabled in order by default. Each group can be configured to have a length of between 1~8 bits and fill in either code or one's complement code.

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

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

Start: A fixed bit is inserted in front of the header and data code.

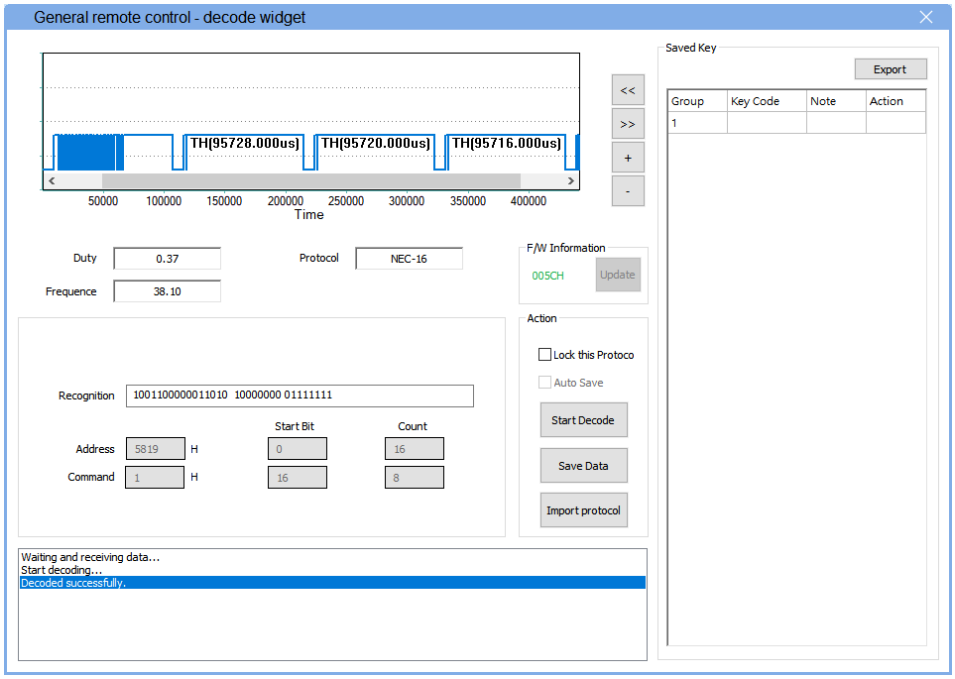
Toggle: A variable bit is inserted to distinguish whether the two codes sent by a remote controller in the full code repeat mode is triggered by a long key press condition or by a two consecutive press conditions.

Sending process: The sending process is shown with a timeline graph. The boot code is sent first, then the constant code, the complement of the constant code, the command code and the complement of the command code, can all be chosen by dragging. The sending order can be adjusted arbitrarily.

2.1.3 Code/Decode Widget

The general remote controller decode widget can be used to assist in the general remote controller development. The decoding function requires connecting the ESK-IRRC-R00 decoding board to the computer for use.

As shown below, the software will draw the sending code waveform using the output method of a general infrared receiver. The MARK with carrier is drawn with a low level and the SPACE without carrier is drawn with a high level. The contents below the waveform diagram are the decoded parameters of the sending code, such as the carrier duty, protocol type and frequency. The decoded 1/0 digit will be automatically filled into the Recognition by the software. Below the recognition data are the decoding Address and Command. 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 to divide the address and command.

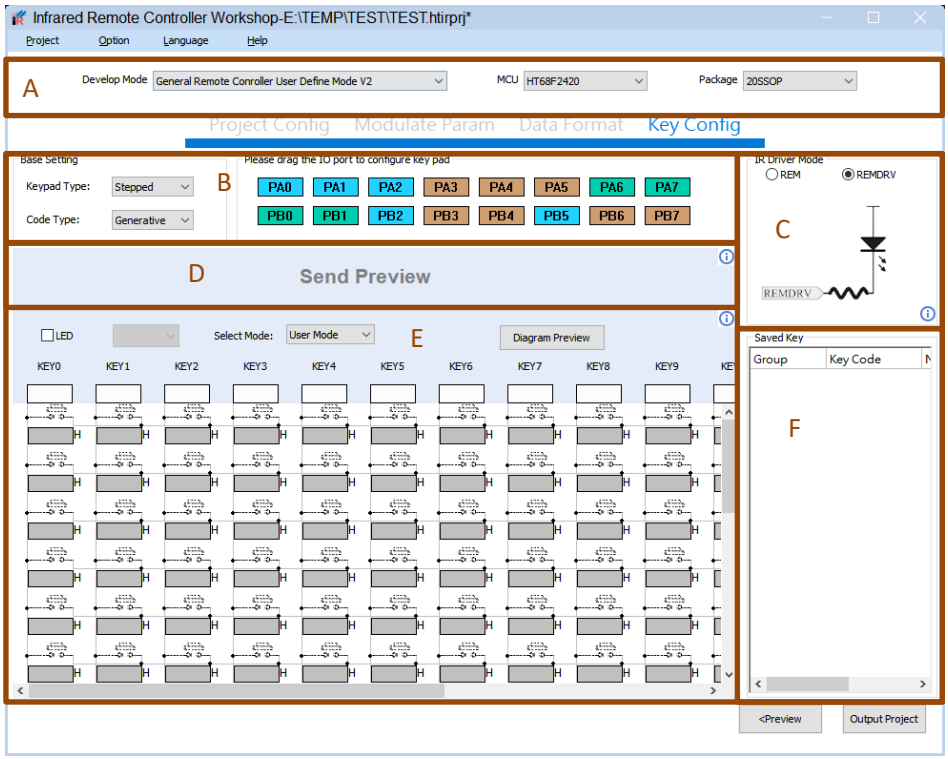


General Remote Controller Decode Widget

After identifying the protocol, the protocol can be locked and multiple keys on the remote controller can be decoded and saved. After saving all the necessary keys for development, pressing the Import Protocol can import the learned data into the remote controller solution under development.

2.1.4 Keys and Driver

The final step in developing a standard or user defined protocol for a general remote controller is key configuration. As shown in the following figure, area A is a solution type and MCU type configuration area, area B is an I/O area for selection and base setting area, area C is an IR LED driver mode selection area, area D is a send preview area, area E is an I/O configured key and command code write area and area F is a saved key area for code development.

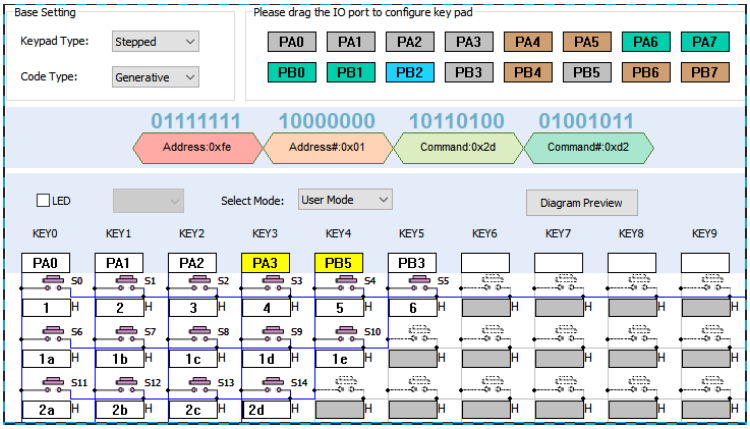


Key Configuration Page

2.1.4.1 Keys and I/Os

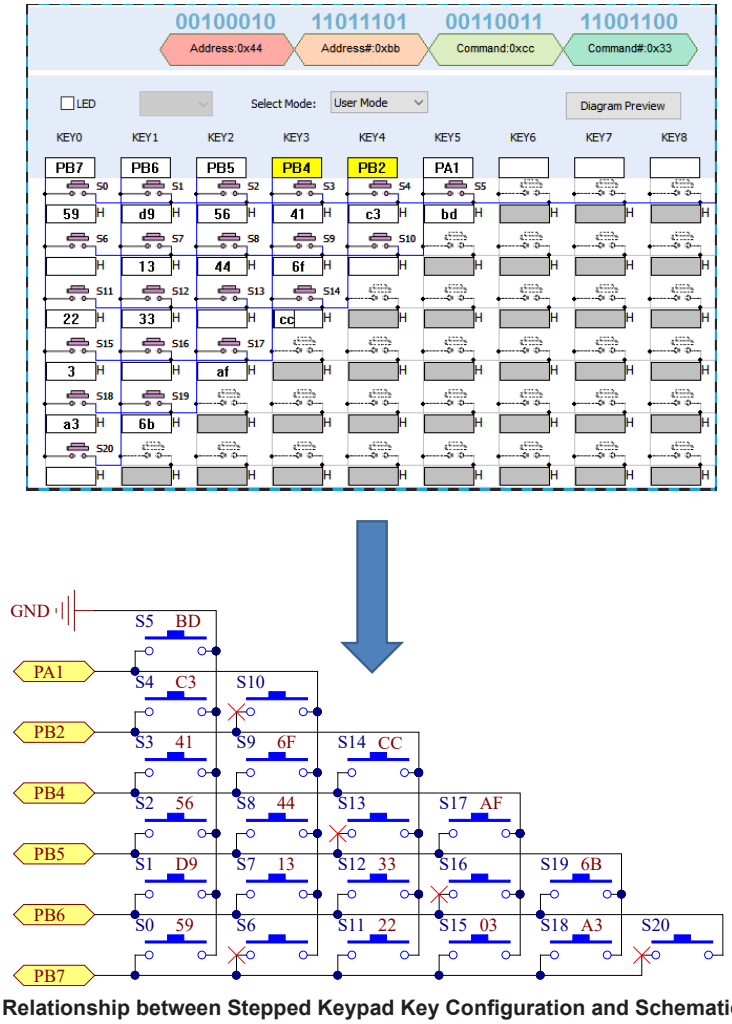
Area A, at the top left of the configuration page, can select the MCU package type. In area B, the base setting can select different keypad type and project output code mode, the blue pin icon in the I/O area for selection is common to all package types of the device, the yellow pin icon is unique to the 16NSOP and 20SSOP/20NSOP packages and the green pin icon is unique to the 20SSOP/20NSOP packages, the gray pin icon represents pins that have been used or are not included in the currently selected package. The I/O icon can be dragged into the key pin position below area E.

2.1.4.2 Stepped Keypad



Stepped Keypad Key Configuration Diagram

As shown above, when the stepped keypad type is selected and the I/O pins are configured to the Key configuration table, a stepped keypad is generated automatically. The maximum number of keys that can be generated is calculated using the formula: K=N×(N+1)/2, where N is the number of the configured I/O pins.



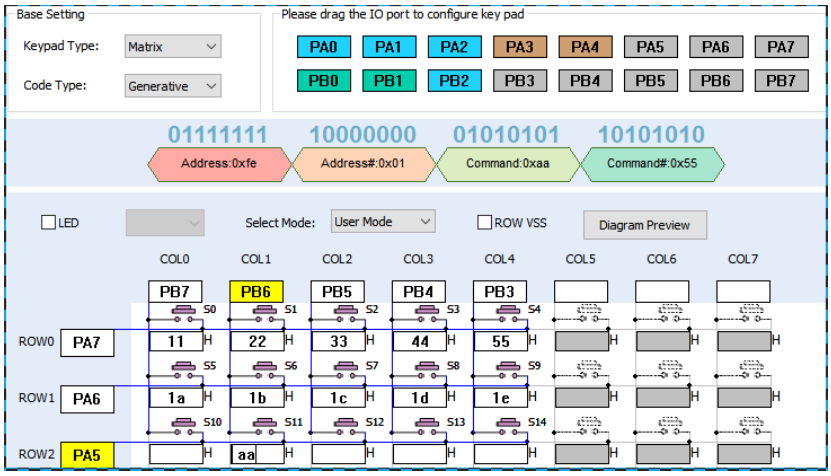
Relationship between Stepped Keypad Key Configuration and Schematic

After the I/O pins are configured to the stepped 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 above upper half, PB4 and PB2 are the two I/Os connected with key S14. After the command code is entered into the corresponding key code table, the send preview area will prompt the sent code in binary. The binary code in the preview is listed from left to right in terms of time. The binary code of the LSB first transmission protocols are opposite to each other.

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.

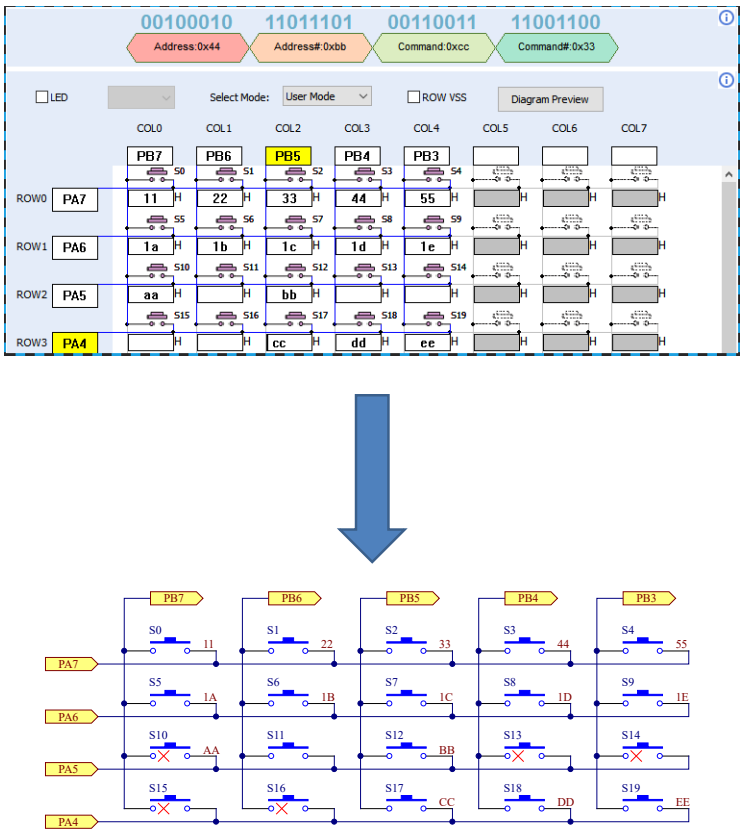
The figure above is a stepped keypad schematic diagram corresponding to the stepped matrix key table below. Keys without configured key values are represented by crosses in the schematic diagram. S6, S10, S13, S16 and S20 in the figure have been included in the scanned code table during configuration 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 keycode table.

2.1.4.3 Matrix Keypad



Matrix Keypad Key Configuration Diagram

As shown above, when the matrix keypad type is selected, after the I/O key configuration table is changed to the matrix mode of ROW and COL, a matrix keypad will generate automatically. The maximum number of keys that can be generated is calculated using the formula: K=ROW×COL. For the 8-pin SOP package, 5 I/O pins can configure up to 6 keys. For the 16-pin NSOP package, 12 I/O pins can configure up to 36 keys. For the 20-pin NSOP/SSOP packages, 16 I/O pins can configure up to 64 keys. The ROW VSS can add additional COL number of keys.



Relationship between Matrix Keypad Key Configuration and Schematic

After the I/O pins are configured to the matrix 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 above upper half, PB5 and PA4 are the two I/Os connected with key S17. After the command code is entered into the corresponding key code table, the send preview area will prompt the sent code in binary.

If the ROW VSS is enabled, when clicking on the command code on the ROW VSS 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.

The figure above is a matrix keypad schematic diagram corresponding to the matrix key table below. Keys without configured key values are represented by crosses in the schematic diagram. Even if the key values have not been configured or if the keys have not been soldered onto the physical circuit, they will be included in the scanned code table during configuration generation. For unconfigured keys, fill in 00H in the software-generated key code table.

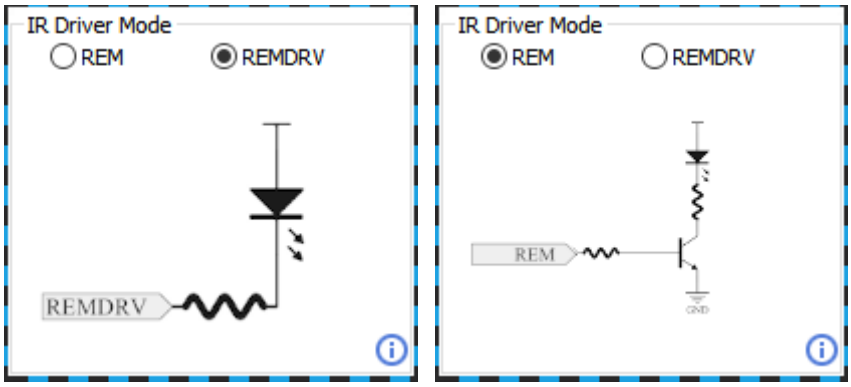
2.1.4.4 Develop Mode

When the HT68F2420 20-pin SSOP MCU package type is selected, the keypad configuration mode can be set to develop mode. The keypad compatible with the development board can be one-click configured, a stepped keypad can be configured with 136 keys, while a matrix keypad can be configured with 72 keys. Because the HT68F2420 ESK-IRRC-T00 development board is drawn in the form of a stepped keypad and printed with silk screen on the PCB. Therefore, the software identifies the matrix keypad key positions according to the silk-printed key positions corresponding to the I/O configuration on the ESK-IRRC-T00 development board, rather than the natural sequence.

2.1.4.5 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 below.



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 a larger drive current to directly drive IR LED. Using the REM driver mode requires to combine with an external bipolar or MOSFET transistor to improve the output drive capability.

The I/O pin to drive an LED indicator can be selected in area E for a functionally independent pin. In the stepped keypad, it can also be pin-shared with KEY0. 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.

Indicator Configuration

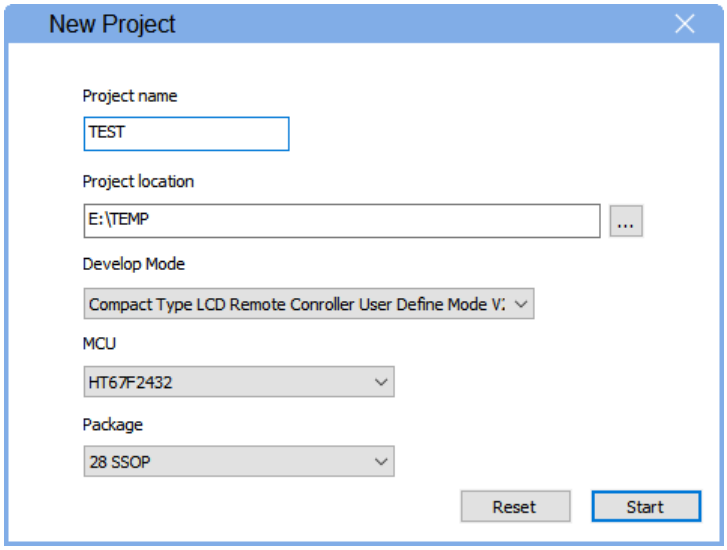
After the key configuration is complete, click circuit diagram preview to generate the remote controller schematic diagram.

After completing the configuration, users can choose the type for the generated code in the Base Setting in area B. Users can choose either the Conditional Compilation or Generative method to generate IDE. The Conditional Compilation is more convenient for secondary development and the Generative programs have better readability.

After completion, click output project 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



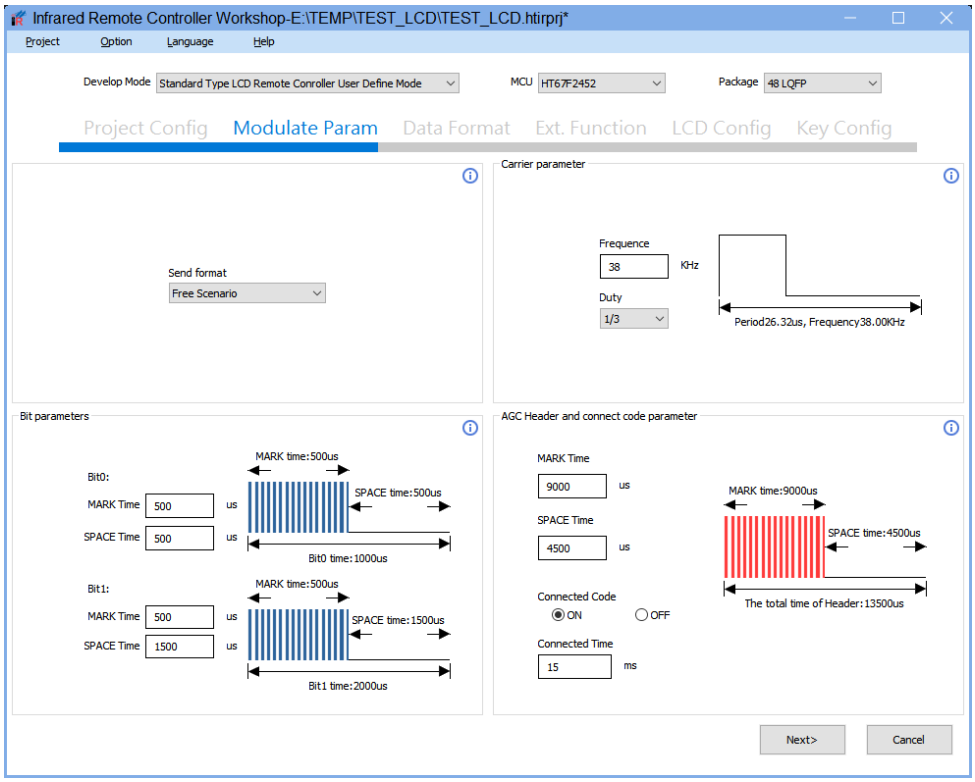
Project Setting for LCD Remote Controllers

According to the different resources of the Holtek LCD remote controller MCUs, the workshop provides two solutions, the Compact Type LCD Remote Controller and the Standard Type LCD Remote Controller. Both solutions have basic key detection, LCD display and infrared emission functions. The Standard Type LCD Remote Controller has more available resources compared to the Compact Type LCD Remote Controller, adding several optional configuration functions. The specific differences between the two solutions are shown below:

	Compact Type LCD Remote Controller	Standard Type LCD Remote Controller
Code Mode	Single-group Single/Double frame	Single-group Single/Double frame Double-group Single/Double frame
Timer Mode	2 options available	3 options available
Double Key Simultaneous Pressing	×	√
Switch Units	×	√
Lock Keypad	×	√
Clock Display	×	√
Applicable ROM	2K	4K and above

2.2.1 Modulation Mode

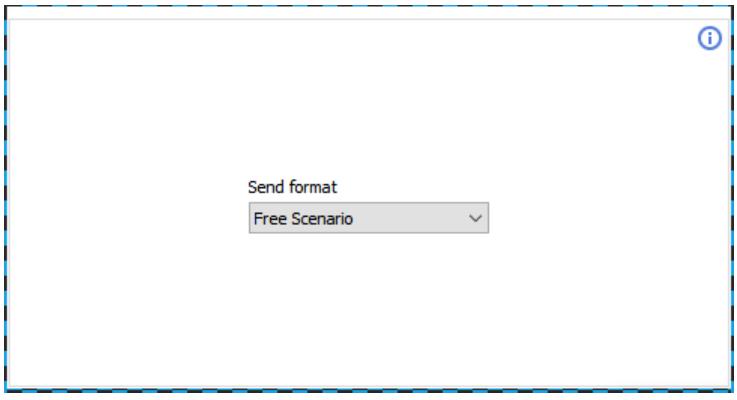
The workshop software provides a user defined protocol mode for LCD remote controller development. Users can define their own parameter and code rules according to their product requirements.



Modulation Mode Configuration Page

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.

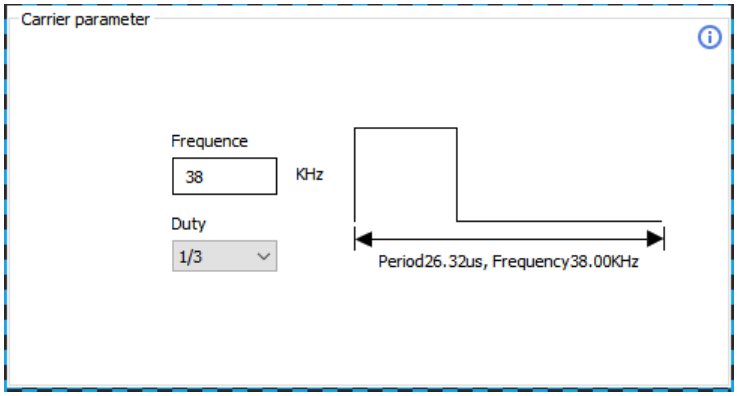
Send Format



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

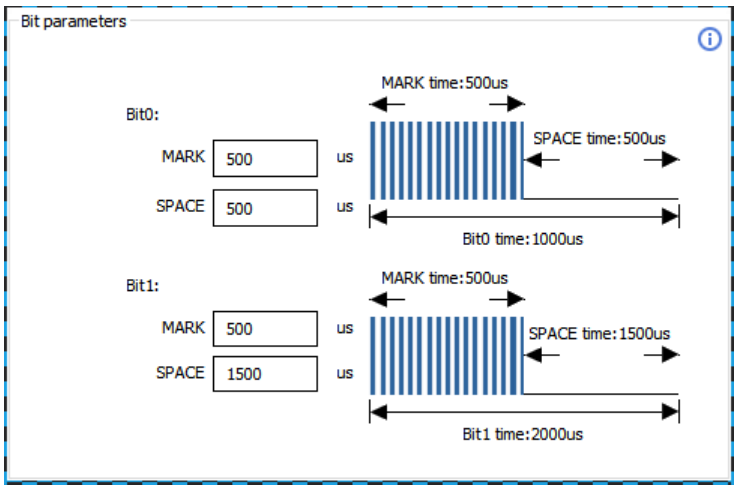


Carrier Parameters

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.

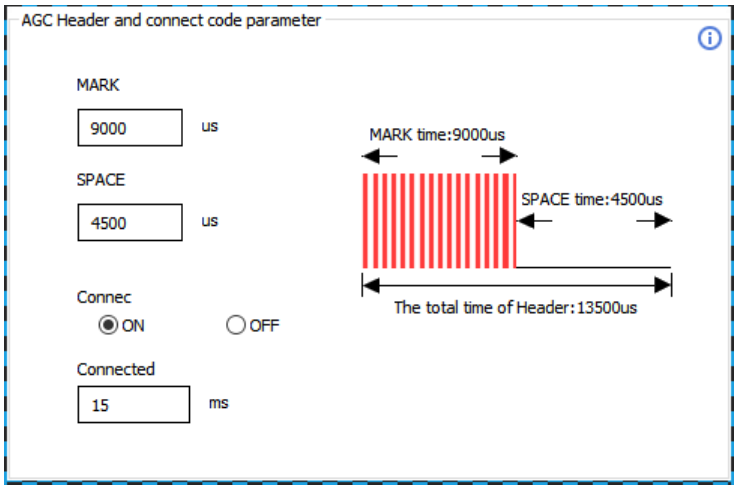


Bit Parameters

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 universal data code. The longer MARK in the AGC Burst type boot code assists the receiver to initialise the gain. The software limits the header time between 3ms and 20ms. The MARK, SPACE and total time parameters are marked graphically by the software.

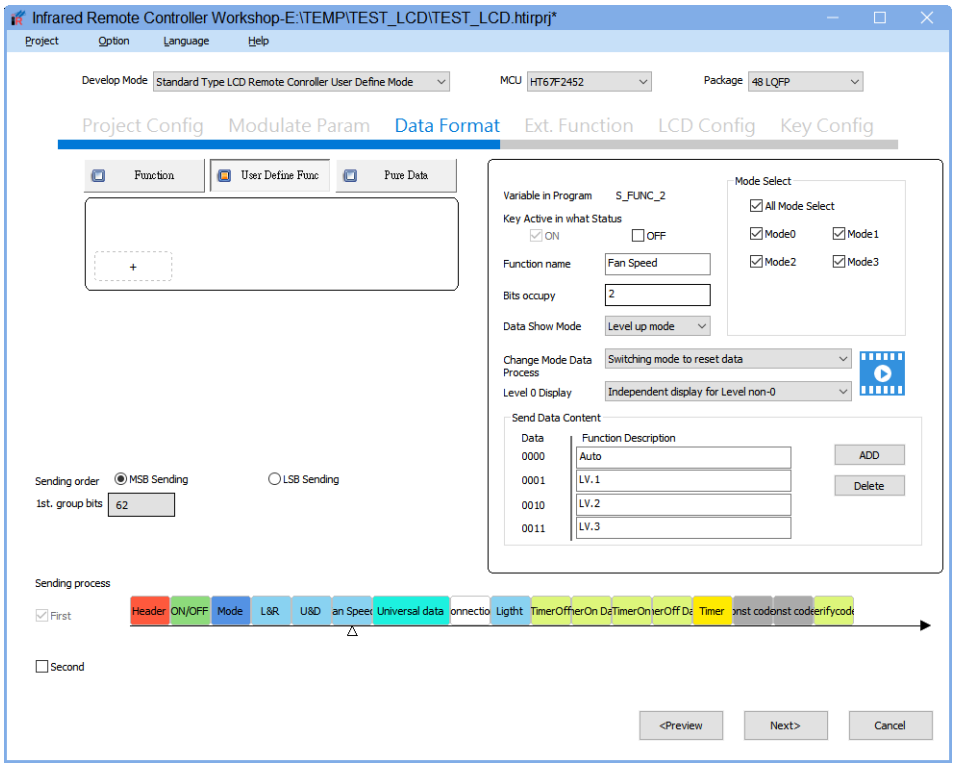
The Connect Code is used to insert a long SPACE in the middle of a long string of bit codes in the LCD remote controller. The connect code can not only be used for the decoding software to identify the protocol but can also be used to avoid the MCU LVR reset problem due to rapid voltage drops caused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



Header and Connect Code Parameters

2.2.2 Data Format

The second step in configuring the code is to configure the data format. The data format for the common LCD remote controllers is to modulate the LCD display content into the sending data.



Data Format Page

The data format page is mainly used to configure the Send Data.

The left side of the page is the configurable list for Send Data, the right side of the page is the configurable content for Send Data. The Sending order in the lower left side can be configured, either LSB or MSB first. The bottom side displays the sending process in a timeline graph. The Compact Type LCD Remote Controller provides one set of data transmission method, while the Standard

Type LCD Remote Controller provides two sets of transmission methods. The double-group mode application can be used through secondary development to send according to conditions. A typical application is for Gree's air conditioner remote controller, which sends two groups only when the timer function is enabled and sends one group only when the timer function is disabled to reduce power consumption.

The list on the left side is divided into three types of data based on functional differences, namely general function data, user defined function data and pure data. To configure the data to be sent, simply requires to select the corresponding data block and drag it to the sending process below.

The general function data includes ON/OFF, Mode and Timer, among which the ON/OFF and Mode are essential functions, and each function has a corresponding LCD display and corresponding keys.

The user defined data function is a sending function that can be configured by users. It can be configured according to user's needs. Each function has a corresponding LCD display and corresponding keys.

The pure data includes universal data, constant code, timer related data, verify code and free data. The universal data has corresponding LCD digital displays controlled by the "+" and "-" keys, while other pure data does not have corresponding LCD displays or keys.

The right side of the page contains user configurable parameters.

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, the user defined functions and free data can be changed to the corresponding names according to development requirements;

Bits Occupy: the number of bits occupied by the currently configured function in the sending process. The minimum occupied bits are configured according to the speed level in binary by default (can be modified during secondary development);

Mode selection (user defined data function configuration): configure user defined functions to be available in specific mode keys;

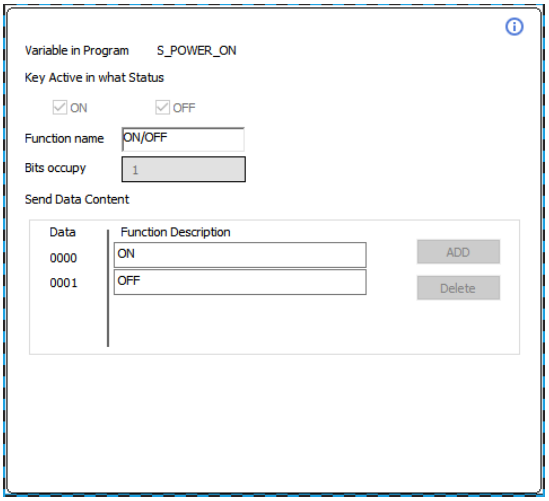
Data Show Mode (user defined data function configuration): Configure LCD display modes associated with the user defined functions;

Change Mode Data Process (user defined data function configuration): the data processing mode for current user defined functions data when presses the "Mode" key at the remote controller to switch modes;

Zero level Display (user define data function configuration): whether to display this function in 0 and non-0;

Send Data Content: configure the level number and function description for user defined functions.

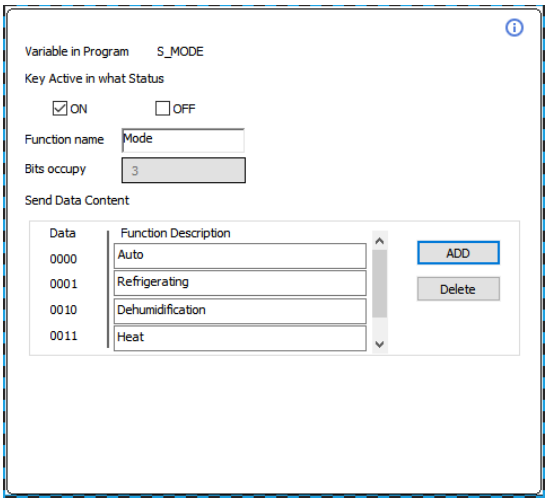
On/Off



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.

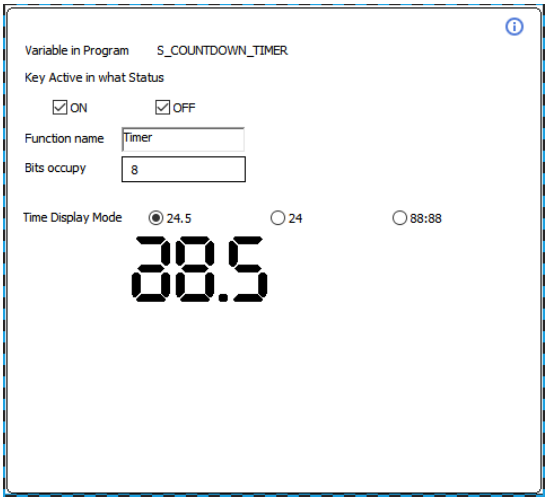
Mode



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 remote controller mode is a switching display.

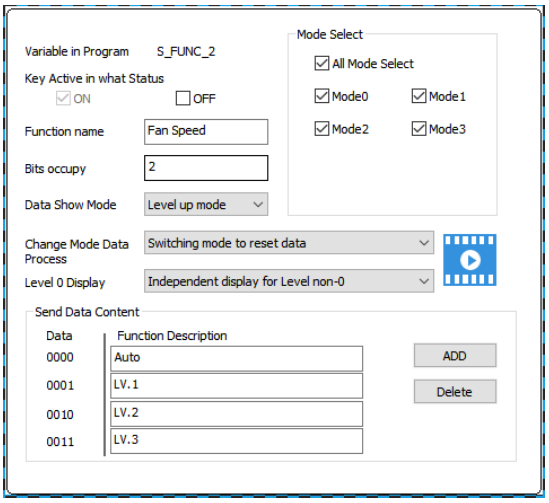
Timer



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 timer mode will automatically exit. The minimum operation time for 24.5 mode is half an hour, for 24 mode it is 1 hour, and for 88:88 mode it is 1 minute. 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.

User Defined Functions



User Defined Functions Configuration

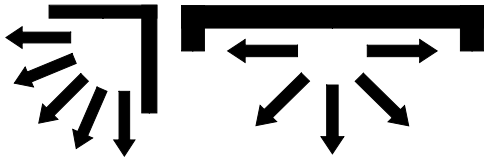
The user defined functions can be configured according to the user required key function characteristics.

The function name can be modified, and the configured LCD and keys will be displayed as the modified custom functions.

According to the remote controller LCD display mode, data can be divided into the following types:

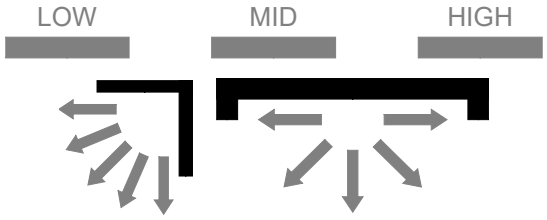
(1) ON/OFF mode: Data displayed in the ON/OFF mode, such as the sweep up and down, sweep

left and right, of the air conditioner remote controller, and the swing function of the fan remote controller. The data definition is 1 for on and 0 for off. As shown in the black icon below.



Common ON/OFF Mode Display Examples

(2) Shift mode: Data displayed in the Shift mode, such as the specified blowing direction for the sweep function of the air conditioner remote controller, and the high, medium and low fan speed of the fan remote controller; the direct blowing, natural and sleep type display modes. As shown in the gray icon below.

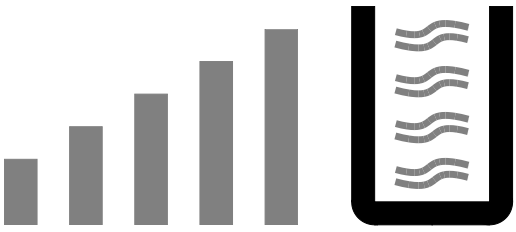


Common Shift Mode Display Examples

(3) Bit mode: Using the carry mode to display data that requires simple addition, which can be used for water storage control of the water heater remote controller or sterilisation time type display control. As shown in the gray icon below.

Common Bit Mode Display Examples

(4) Level up mode: The display mode that adds one block for each level up. Generally used for the air conditioner remote controller fan speed level control or water heater water level control. As shown in the gray icon below.



Common Level up Mode Display Examples

(5) Digital mode: Display the function level in digits.

There are three methods for Change Mode Data Process: Shared data, Independent data, and Switching mode to reset data.

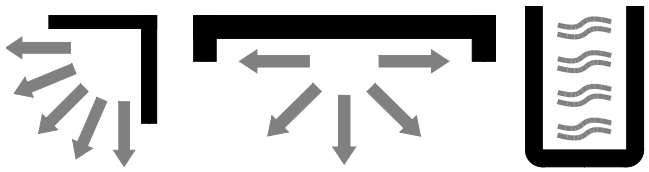
When the mode key is pressed to perform the mode changing operation, the differences in the use of the three data processing methods are as follows:

(1) Sharing data will keep the function level or ON/OFF status unchanged. For example, the air conditioner remote controller will still maintain the ventilation function after the mode is changed.

(2) Switching mode data is not reset: Independent data will allocate independent cache for each mode of this function. Switching mode will automatically load the corresponding level of the previous mode. For example, the air conditioner remote controller fan speed function cooling mode is set to 3rd level. After switching to other modes and changing levels, it can still maintain 3rd level when switching to cooling mode.

(3) Switching mode to reset data: which will reset the function level to zero. For example, when the air conditioner remote controller is in the super cooling function status, it will ensure that the super cooling function is immediately turned off when the air conditioner is switched from cooling mode to non-cooling mode.

Level 0 is not displayed on the LCD by default. Some devices require a special display for Level 0 (such as “AUTO”), and the zero level display can be enabled in the user defined function configuration. Some devices require a non-0 status to display working (such as the black display of the icon below), then the non-0 status display can be enabled on this user defined function configuration.

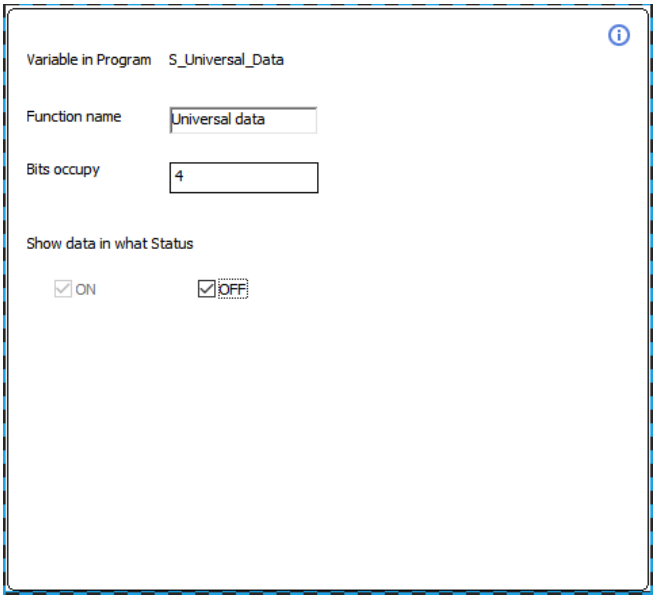


Common Level Non-0 Display Examples

The send content design can increase or decrease the level of the current design function. The maximum level for the ON/OFF mode data is 1 level, for Shift mode is 7 levels, for Bit mode is 15 levels, for Level up mode is 7 levels and for Digital mode is 9 levels.

Pure Data

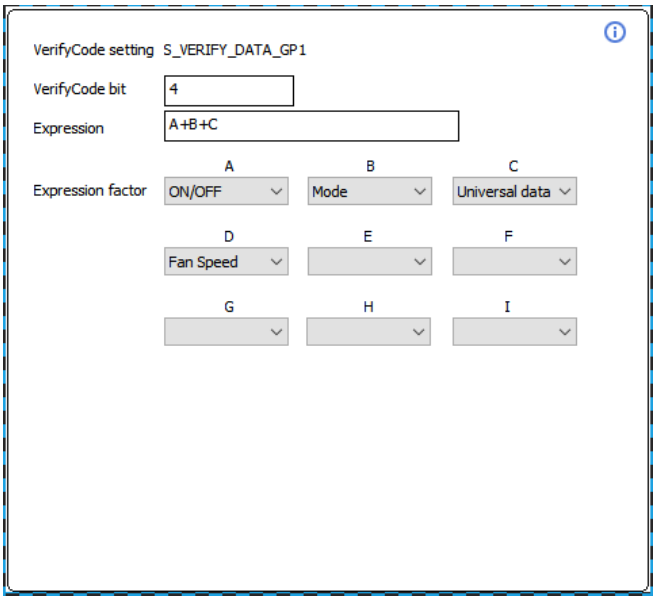
- **Universal Data**



Universal Data Configuration

The universal data is the data used by the LCD remote controller for global digital display. The common applications include the air conditioner remote controller temperature, stepless fan remote controller speed level, water heater/bathroom ceiling heater and ventilator remote controller temperature and the lighting product light intensity rang. The changes in universal data are controlled by the "+" and "-" keys on the remote controller.

- **Verify Code, Verify2**

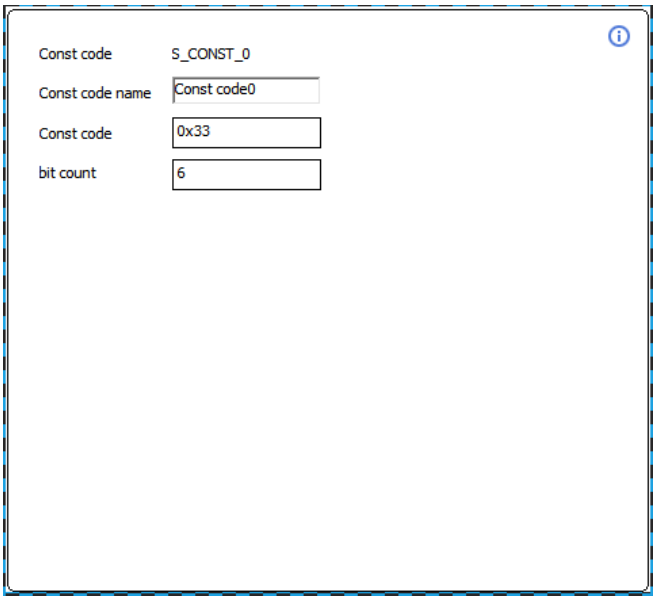


Verify Code Configuration

The verify code in the code is used for verification. 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. The compact type LCD remote controller has less ROM and RAM resources, it is not recommended to perform multiplication, division or other complex operations for the verify code.

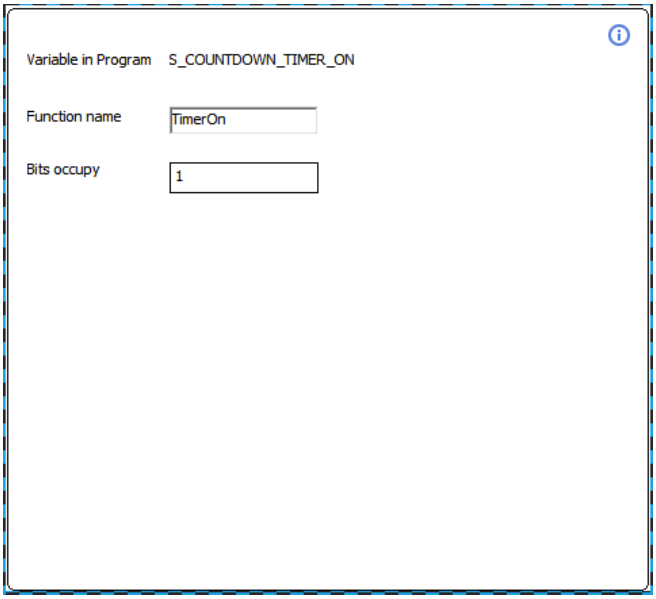
- **Constant Code**



Constant Code Configuration

The constant code is used for verification which has a pure data content and can be configured up to 10 sets. 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.

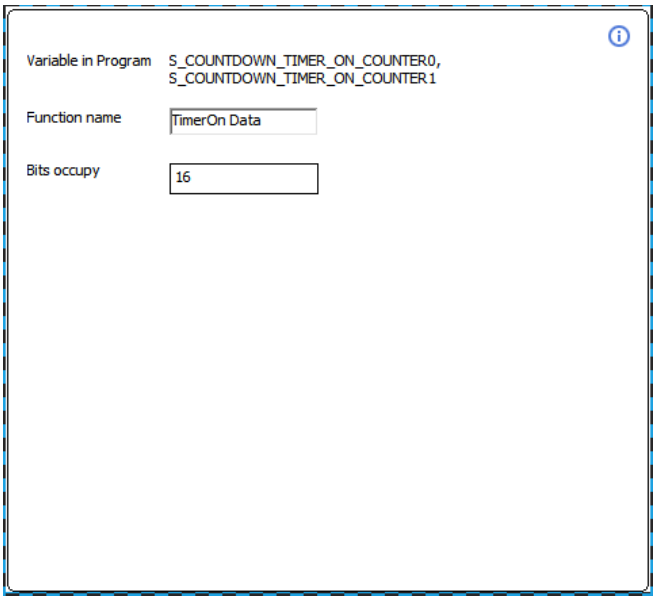
- **Timer On, Timer Off**



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**



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. The timer on counter and timer off counter functions correspond to two 16-bit variables. When the timer function display mode is set to 24.5 or 24 mode, the timer on counter and timer off counter registers store the total remaining minutes, and it is recommended to set the bit length to 11~16 bits. When the timer function display mode is set to Mode 3, the default lower 8 bits are for minutes, the higher 8 bits are hours, and the bit length is fixed at 16 bits.

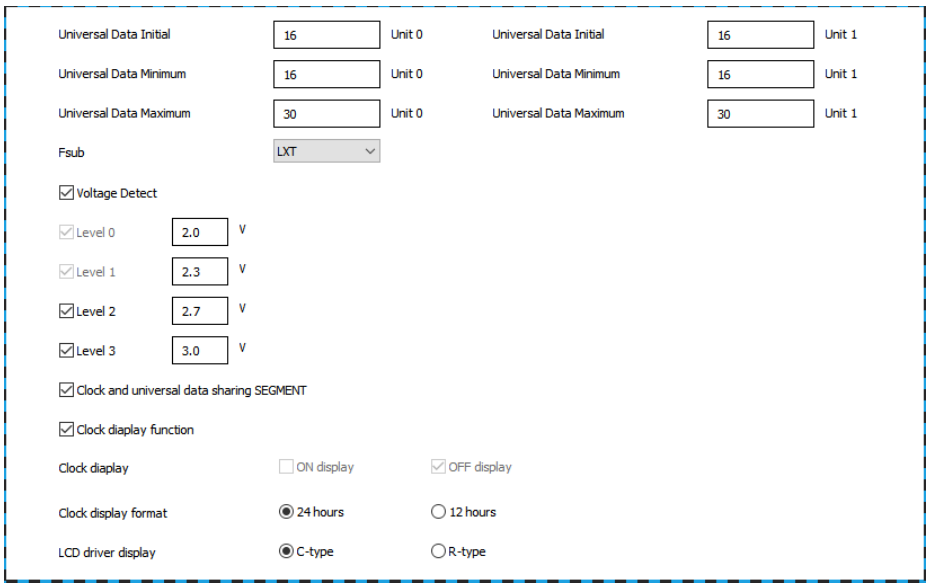
If the user has set a timer on or off counter using the remote controller and then uses the remote controller to send other functions (such as Level up or Shift modes) within half an hour or one hour, if the remote control data sent again remains unchanged, the timer time will be refreshed. Therefore, by adding the power-on/power-off timer to the sent data and sending the data with the remaining minute data calculated by the remote controller MCU, can ensures that the previously set power-on/power-off timer for the device remains unchanged.

- **Free Data**

The free data is a set of variable data in the sending process. For example, after either the sweep up and down or the sweep left and right functions of Gree's air conditioner remote controller has been enabled, one of the bits in the remote controller's sending process will be set to 1 for confirmation.

The free data of the workshop only generates one data variable in sending process, and the default status of the data is 0. The data transformation operation requires to be implemented by the user through secondary development after the program is produced. The free data can be set to a length of 1~8 bits.

2.2.3 Function Configuration



Send Data Configuration

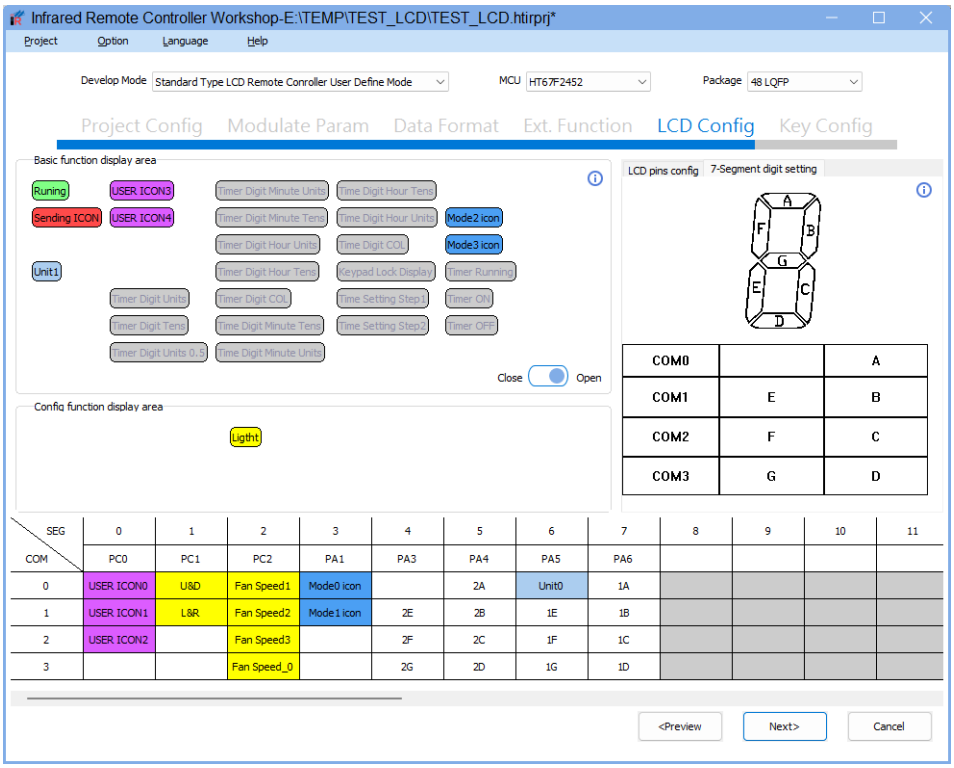
The universal data can display data ranging from 0 to 99, and its range can be limited in the functional configuration. For example, most air conditioners have a temperature range of 16℃ to 30℃, while most water heaters have a temperature range of 30℃ to 80℃. The Compact Type Remote Controller provides one unit of display data, while the Standard Type Remote Controller provides two units of display data.

The Voltage Detect function is implemented by detecting the battery voltage, which can be configured in 2~4 levels. After enabling the Voltage Detect, when the voltage is detected to be lower than the minimum level, the remote controller will not perform IR emission function.

The Clock display supports optional display of either 12 hours or 24 hours.

2.2.4 LCD Configuration

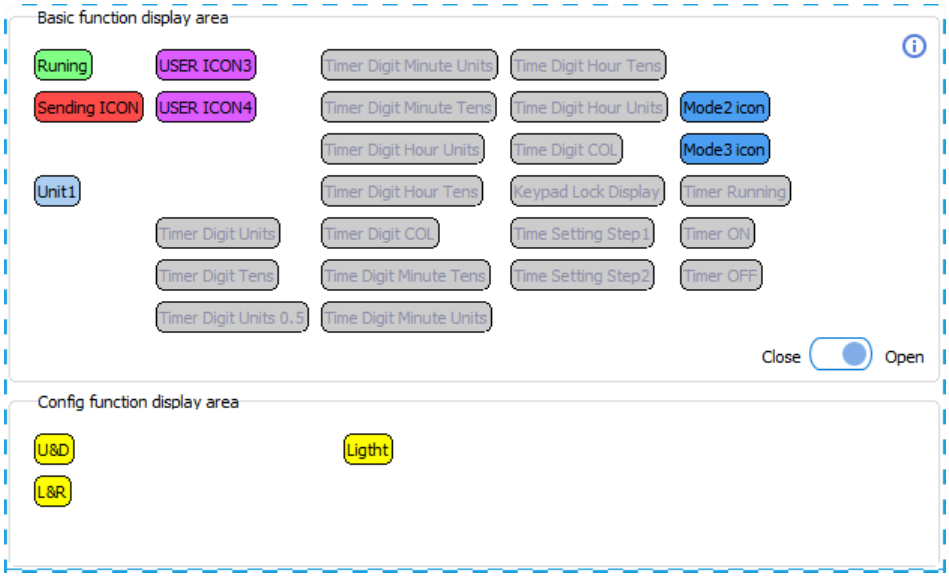
The workshop assists users in developing the LCD display function in the form of LCD RAM corresponding to LCD panel blueprint tables. The LCD remote controller only supports LCD drivers in 1/4 duty and 1/3 bias mode.



LCD Display Overall Configuration

The top left of the LCD configuration page is an LCD content area to be configured, which is divided into two parts: the basic function configuration area and the user defined function configuration area.

LCD Display Configuration Area



LCD Display Configuration Area

The display segments corresponding to the enabled functions in the area to be configured are drawn in colour and can be dragged to the COM/SEG table configuration area below. The display segments corresponding to the disabled functions are drawn in gray and cannot be used. The user defined

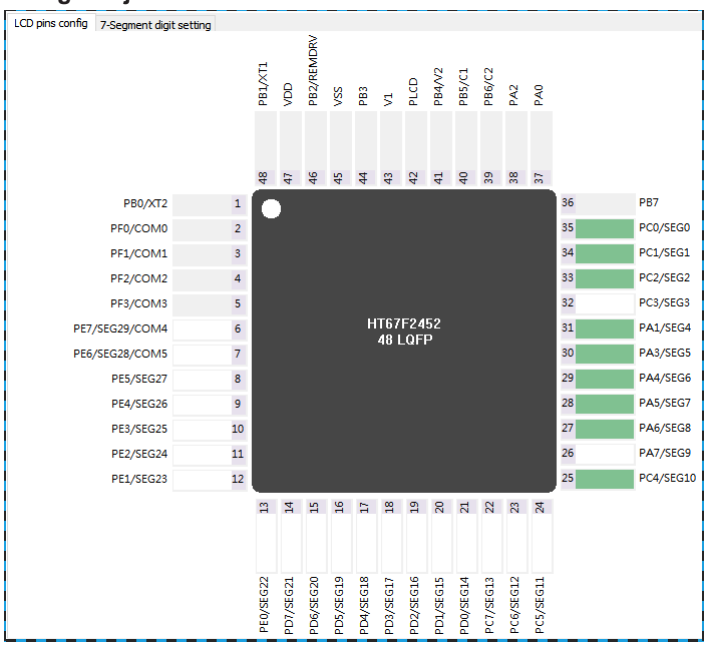
blocks will automatically light up when the power on is pressed, and will not light up when power off. If the user requires to make related display actions according to changes in other functions, the user can change the user defined display operation in the secondary development after the project is output.

The user defined function display area generates the relevant display blocks that can be dragged based on the user defined functions and the data format of each function.

Drag the enabled display block to the RAM Config function display area below to enable the corresponding display function in the remote controller.

The switch of Close and Open allows user to view the segments displayed in the power-on and power-off status.

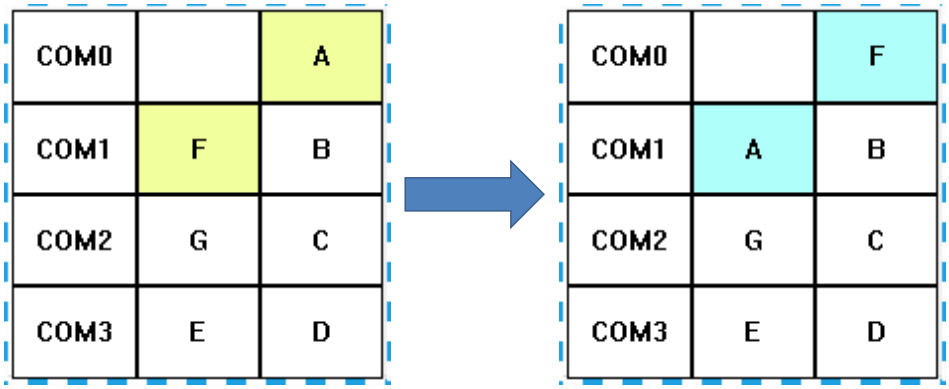
LCD Pins and Digit Adjustment Area



LCD Pins and Digit Adjustment Area

The right side is LCD pins and digit adjustment area. The SEG pins for the MCU to drive an driver LCD can be configured on the LCD pin configuration page. In the LCD pin configuration area, the default COM pins configuration is 4-COM mode. The SEG pins can be enabled by clicking on the white pin box, and the pin will turn green after enabled.

The 7-Segment digit setting page can configure the digital display. The software supports four COMs and two SEGs to form an A~G 7-segment digit, and can be compatible with different LCD screens by adjusting the segment arrangement in the digit adjustment area. 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.



Exchange Method in LCD Digit Adjustment Area

RAM Mapping Area

SEG / COM	0	1	2	3	4	5	6	7	8	9	10	11
	PC0	PC1	PC2	PA1	PA3	PA4	PA5	PA6	PC4			
0	USER ICON0		Fan Speed1	Mode0 icon		2A	Unit0	1A	BAT cell-0th			
1	USER ICON1		Fan Speed2	Mode1 icon	2E	2B	1E	1B	BAT cell-1st			
2	USER ICON2		Fan Speed3	Mode2 icon	2F	2C	1F	1C	BAT cell-2nd			
3			Fan Speed_...	Mode3 icon	2G	2D	1G	1D				

LCD RAM Relational Table

The left column of the RAM mapping area contains 4 COMs. The top row are the SEG pin configuration. The pins enabled as SEG pins will automatically configured from left to right to the SEG column. The configured I/O can be adjusted by dragging.

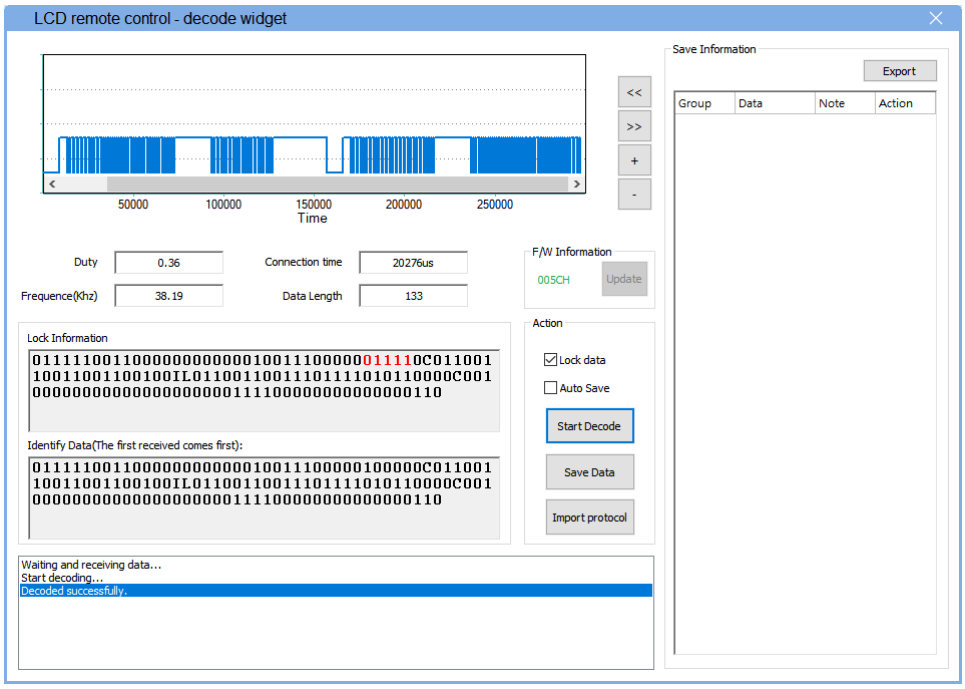
The digital display consists of 7 segments. Dragging the digits to the RAM area, the software will automatically allocates two consecutive SEGs and occupies 7 bits of space in the LCD RAM. For digits that have been allocated, the RAM bits occupied by the display digits can be dragged to other free RAM positions.

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.

2.2.5 Code/Decode Widget

The LCD remote controller decode widget can be used to assist in the LCD remote controller development. The decoding function requires connecting the ESK-IRRC-R00 decoder board to the computer for use.

As shown below, the software will draw the sending code waveform using the output method of a typical infrared receiver. The MARK with carrier is drawn with a low level and the SPACE without carrier is drawn with a high level. The contents below the waveform diagram are the decoded parameters of the sending code, such as the carrier duty, protocol type and frequency. The decoder 1/0 digit will be automatically filled into the Recognition by the software. Checking the lock data in the action area can add the decoded data to the lock area. When decoding the next key, the recognised data will be automatically compared with the locked data. If there are differences between the new data and the locked data, the software will prompt the corresponding data in red to indicate changes.



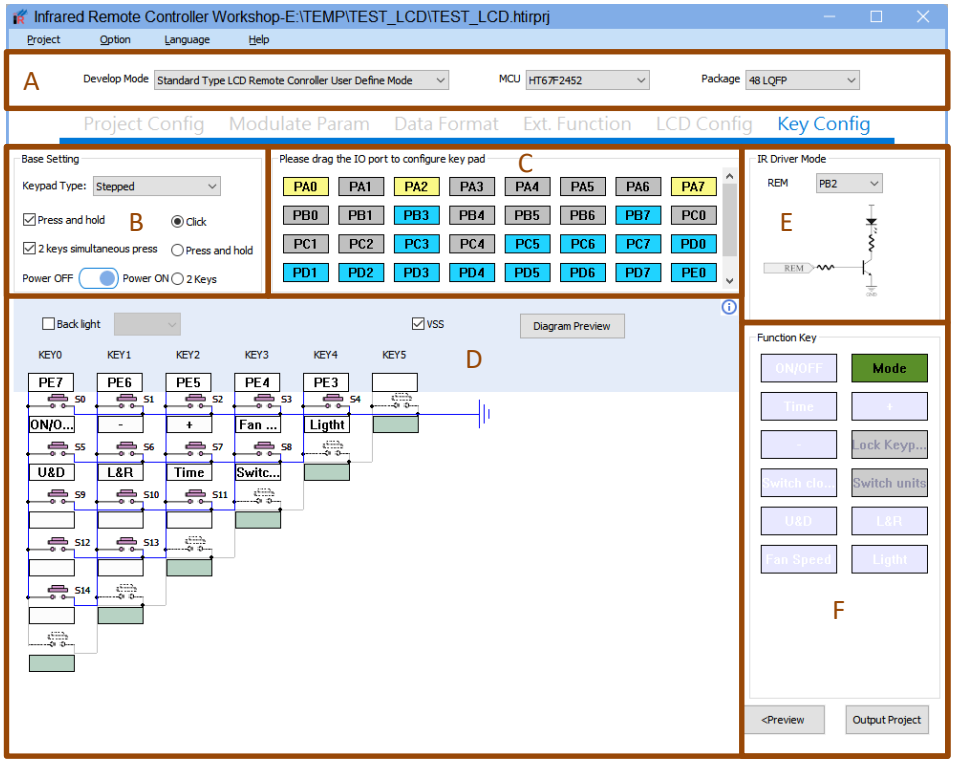
LCD Remote Controller Code Mode

The LCD remote controller code/decode is only used to learn the LCD remote controller PDM mode code. By default, codes with longer SPACE are set to 1, while codes with shorter SPACE are recognized as 0.

The import protocol data function is only used as a data record for code. For the convenience of users to record and compare parameters, it does not involve data format or key configuration.

2.2.6 Keys and Driver

The final step in developing an LDC remote controller is key configuration. As shown in the following figure, area A is a solution type and MCU type configuration area, area B is an base setting area, area C is an I/O area for selection, area D is a key action configuration area, area E is an IR LED drive configuration area and area F is a function area for selection.



Key Configuration Area

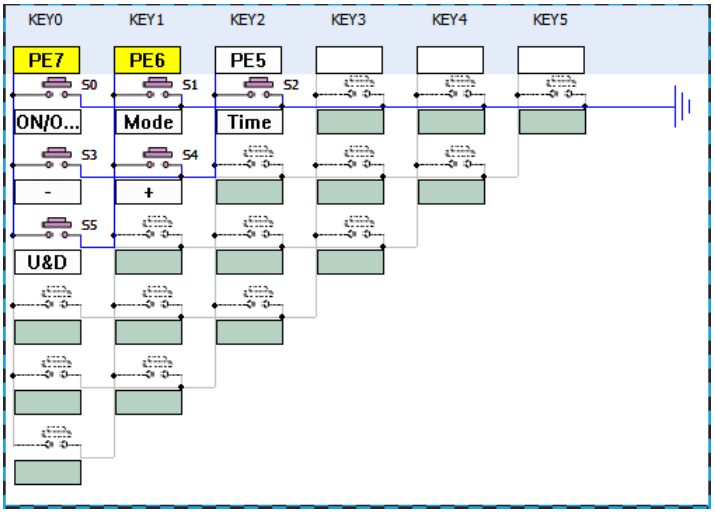
2.2.6.1 Keys and Functions

Area A, at the top of the configuration page is MCU information area, can change the remote controller MCU package type. Area B is a base setting area, can select a stepped keypad or matrix keypad, and can configure whether the long key pressing or double key simultaneous pressing is enabled for the keypad. Area C is an I/O area for selection, the gray blocks indicate used or occupied I/Os. Area D is an I/O configuration area, the I/O icon in area C can be dragged into the I/O configuration area in area D. The back light function of area D is for the back light of the keypad or display screen, which will light up for 4s after pressing any key. After the I/O is configured to the keypad, configurable function bits can be enabled in area E. Drag the key function in area F into the key function block to enable the function.

The ON/OFF status switching key in area D can switch the view and configure the ON or OFF status configuration function.

The main functions of the remote controller are configured to operate in short key pressing way, while a few functions (such as continuous addition and subtraction, setting the clock, locking the keypad and switching units) are implemented by long key pressing or double key simultaneous pressing.

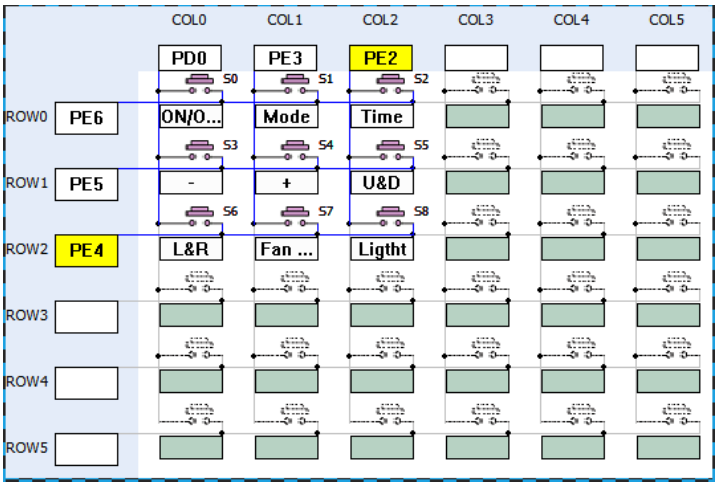
2.2.6.2 Stepped Keypad



LCD Remote Controller Stepped Key

As shown above, when the stepped keypad type is selected and the I/O pins are configured to the Key configuration table, a stepped keypad is generated automatically. When VSS is checked to add to the circuit, the keypad will have the maximum number of keys. The number of keys that can be generated is calculated using the formula: K=N×(N+1)/2. When VSS is not checked to add to the circuit, the double key simultaneous pressing is supported. The number of keys that can be generated is calculated using the formula: K=N×(N-1)/2, where N is the number of the configured I/O pins.

2.2.6.3 Matrix Keypad



LCD Remote Controller Matrix Key

As shown above, when the matrix keypad type is selected, after the I/O key configuration table is changed to the matrix mode of ROW and COL, a matrix keypad will generate automatically. The maximum number of keys that can be generated is calculated using the formula: K=ROW×COL.

2.2.6.4 IR Driver

The IR driver is configured according to the MCU function. If the MCU has built-in IR dedicated driver pins REM/REMDRV, the software will be forced to use the dedicated driver pins to drive the IR LED. If the MCU does not have IR specific driver pins, any PTM output pin can be selected to output the IR drive waveform in REM mode.

3. Workshop Hardware

The available components for the IR remote controller workshop supporting development board are as follows:

Development Board/ Development Board Module	Supported MCU	Function/Application	Supported Development Mode
*ESK-IRRC-T00	HT68R2420	136-key stepped integrated development board	General Remote Controller
*ESK-IRRC-T01	HT67F2432	15-key stepped integrated development board	Compact Type LCD Remote Controller
ESK-IRRC-R00	—	IR remote controller decoder board	All modes decode
ESK-IRRC-TMT00	HT68R2420	Modular development board with IR transmitter	General Remote Controller
ESK-IRRC-TM00	HT67F370	Modular development board	Standard Type LCD Remote Controller
ESK-IRRC-TM01	HT69F340	Modular development board	Standard Type LCD Remote Controller
ESK-IRRC-TM02	HT69F350	Modular development board	Standard Type LCD Remote Controller
ESK-IRRC-TM03	HT69F360	Modular development board	Standard Type LCD Remote Controller
ESK-IRRC-TM04	HT67F2452	Modular development board	Standard Type LCD Remote Controller
ESK-IRRC-TMK00	BS67F2432	Modular development board with touch key	Compact Type LCD Remote Controller
ESK-IRRC-TDT00	—	Modular development board with display and IR transmitter	Compact Type LCD Remote Controller Standard Type LCD Remote Controller
ESK-IRRC-TK00	—	24-key matrix keypad modular development board	General Remote Controller Compact Type LCD Remote Controller Standard Type LCD Remote Controller
ESK-IRRC-TK01	—	136-key stepped keypad modular development board	General Remote Controller Compact Type LCD Remote Controller Standard Type LCD Remote Controller

Note: *The on board chip of the integrated development board uses a real MCU and does not support the emulation function.

3.1 Remote Controller Development Board

There are two development board types for IR remote controller workshop, namely integrated development board and modular development board. The integrated development board integrates MCU, transmitter, keys and display (only for the LCD remote controller development board) on one PCB. The modular development board divides the master controller, transmitter, keys and display (only for the LCD remote controller development board) functions of each MCU into several different module boards, which can be assembled by users according to the functional requirements of the development project before development.

3.1.1 Integrated Development Board

The integrated development board ESK-IRRC-T00 is a general remote controller development board using the HT68F2420 as the master controller. It uses 16 I/O pins to form a 136-key stepped matrix keypad, which can emulate any I/O combination of keys.

The integrated development board ESK-IRRC-T01 is an LCD remote controller development board using the HT67F2432 as the master controller. It uses 5 I/O pins to form a 15-key stepped matrix keypad and 20 I/O pins to configure a 4×16 LCD driver.

The specific circuit can be opened in the software options.

Integrated Development Board

3.1.2 Modular Development Board

The modular development board can select different motherboards, combined with the functions of emission, display, and keys required by the remote controller, a complete remote controller solution can be formed. The specific combined solutions are as follows:

Development Type	Motherboard	Development Combined Components 1	Development Combined Components 2
General Remote Controller	ESK-IRRC-TMT00	ESK-IRRC-TK00 ESK-IRRC-TK01	
Standard Type LCD Remote Controller	ESK-IRRC-TM00 ESK-IRRC-TM01 ESK-IRRC-TM02 ESK-IRRC-TM03 ESK-IRRC-TM04	ESK-IRRC-TK00 ESK-IRRC-TK01	ESK-IRRC-TDT00
Compact Type LCD Remote Controller	ESK-IRRC-TMK00	ESK-IRRC-TDT00	

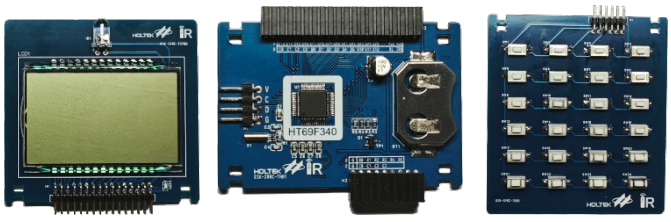
Refer to the circuit diagram for the specific development board connector connection.

The modular development board divides each replaceable function into three parts: display, master controller and keys according to the general remote controller applications.

The general remote controller does not include a display, and can be developed by combining the master controller board and keypad.

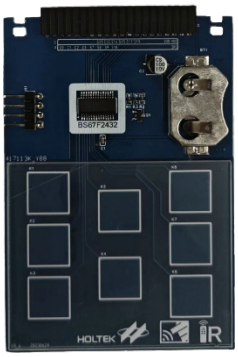
Modular General Remote Controller Development Board

The development for an LCD remote controller requires combining the display board, master controller board and keypad.



Modular LCD Remote Controller Development Board

The BS67F2432 is a touch remote controller MCU. To ensure better touch debugging, the development board ESK-IRRC-TMK00 integrates the touch keys with the master controller on the same board.



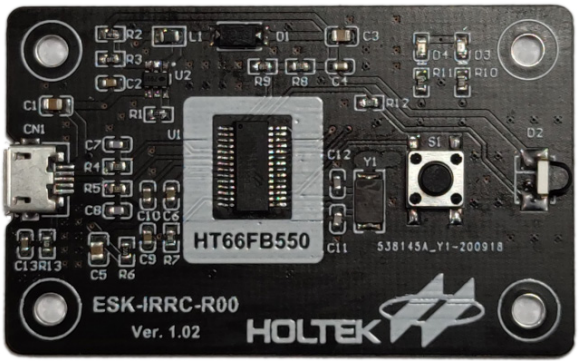
Modular LCD Remote Controller Development Board ESK-IRRC-TMK00

The integrated development board uses an MCU device without the emulation function, while the modular development board uses an EV chip. It should be noted that the EV chip cannot use the PA0 and PA2 I/O pins during emulation.

For the standard type LCD remote controller development board, the circuit uses C type driver by default. Modify the short-circuit jumper on the back of the development board can configure LCD screens with different voltages. Refer to the LCD section of the MCU datasheet for the specific relationship between the short-circuit connection and the C-type bias voltage.

3.2 Decoder Board

The ESK-IRRC-R00 is a decoder board captures infrared carrier signals and uploads them to a computer via a 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.

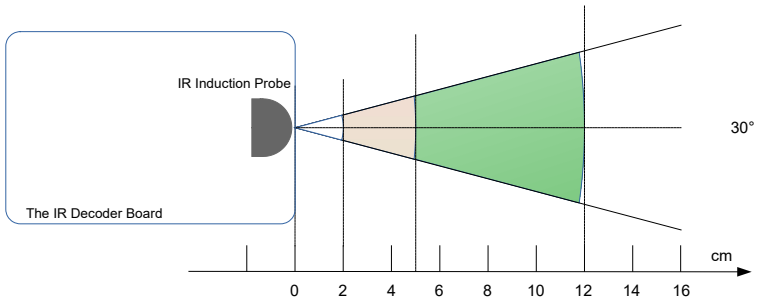


ESK-IRRC-R00 Decoder Board Hardware Diagram

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.

After starting the decoding widget from the software, press the decode key on the software or the key on the decoder board, the LED on the decoder board will change to a constant red light , indicating the decoding starts.

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



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. Description of Other Functions

4.1 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 specially optimised.

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 controller 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.

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 125ms to scan the keys. The LCD remote controller is woken up by a time base to detect the keypad every approximately 125ms, to change the timing/clock digital flashing action every 0.5s, to execute the backlight and clock timing every 1s, to calculate the timer every 4s and to detect the battery voltage every 16s.

4.2 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.

5. Appendix & FAQ

5.1 LCD Remote Controller Functional Data, Keys, Display and Function Correlation Table

Function	Data variables	Associated Keys	Associated Display Variables	Function Definition
Power on/off	b_Power_ON S_POWER_ON	ON/OFF	DIS_POWER	FUNC_POWER_ SWITCH
Mode	S_MODE	Mode	DIS_MODE_x(x = 0~7)	FUNC_MODE_SWITCH
Universal Data	B_Universal_Data S_Universal_Data	+ / -	NUMBER_UNIVERSAL_ DATA_(a)_RAM(b) (a = 0, 1; b = 0, 1)	FUNC_KEY_INC, FUNC_KEY_DEC, FUNC_MODE_SWITCH, FUNC_UNIT_SHIFT_ SWITCH
Timer (mode 1,2)	REG_Countdown_Timer COUNTDOWN_TIMER_ SHOWONESTEP S_COUNTDOWN_TIMER S_COUNTDOWN_TIMER_ OFF S_COUNTDOWN_TIMER_ ON S_COUNTDOWN_TIMER_ ON_COUNTER(a) S_COUNTDOWN_TIMER_ OFF_COUNTER(a) (a = 0, 1)	Timer /ON/ OFF / + / -	DIS_COUNTDOWN_TIMER_ PWRON DIS_COUNTDOWN_TIMER_ PWROFF NUMBER_TIMER_(a)_ RAM(b) (a = 0, 1; b = 0, 1) DIS_TIMER_HALFHOUR	FUNC_COUNTDOWN_ TIMER_SWITCH FUNC_KEY_INC FUNC_KEY_DEC
Timer (Mode 3)	REG_Countdown_Timer_ MINUTE REG_Countdown_Timer_ HOUR S_COUNTDOWN_TIMER S_COUNTDOWN_TIMER_ OFF S_COUNTDOWN_TIMER_ ON S_COUNTDOWN_TIMER_ ON_COUNTER(a) S_COUNTDOWN_TIMER_ OFF_COUNTER(a) (a = 0, 1)	Timer / ON/ OFF / + / -	DIS_COUNTDOWN_TIMER_ PWRON DIS_COUNTDOWN_TIMER_ PWROFF NUMBER_TIMER_HOUR_ (a)_RAM(b) NUMBER_TIMER_MINUTE_ (a)_RAM(b) (a = 0, 1; b = 0, 1) DIS_TIMER_COL	FUNC_COUNTDOWN_ TIMER_SWITCH FUNC_KEY_INC FUNC_KEY_DEC
User Defined Function	ST_S_FUNC_x REG_S_FUNC_x S_FUNC_x (x = 0~19)	User Defined Function / Mode	DIS_MODE_S_FUNC_x DIS_FUNC_x_z DIS_FUNC_x_nz (x = 0~19)	FUNC_x_SWITCH (x = 0~19)

Function	Data variables	Associated Keys	Associated Display Variables	Function Definition
Set Clock	B_TIME_SET_STEP B_TIME_Hour B_TIME_Minute B_TIME_SET_TEMP_HOUR B_TIME_SET_TEMP_MINUTE	Setting / Switch / + / -	NUMBER_TIME_MINUTE_(a)_RAM(b) NUMBER_TIME_HOUR_(a)_RAM(b) (a = 0, 1; b = 0, 1) DIS_TIME_AM DIS_TIME_PM DIS_TIME_SETTING_STEP1 DIS_TIME_SETTING_STEP2	FUNC_TIME_SET_SWITCH ENTER_TIME_SET FUNC_KEY_INC FUNC_KEY_DEC
Clock	B_TIME_Hour B_TIME_Minute B_TIME_Second	None	NUMBER_TIME_MINUTE_(a)_RAM(b) NUMBER_TIME_HOUR_(a)_RAM(b) (a = 0, 1; b = 0, 1) DIS_TIME_AM DIS_TIME_PM DIS_TIME_COL	Time_Scond_plus
Key Lock	b_KEY_LOCK	Key Lock	DIS_LOCK	FUNC_LOCK_SWITCH
Unit	B_UNITSHOW B_Universal_Data	Unit	DIS_UNIT0 DIS_UNIT1	FUNC_UNIT_SHIFT_SWITCH

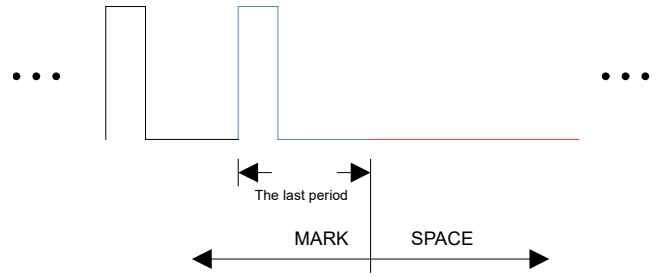
5.2 Special Modulation Code Description and Applications

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.	Because only one bit-length pattern is recognised after decoding, the decoding function configures it as 1.
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 SPACE 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.

5.3 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.



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 above. 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.