Gate control using Remote

INTRODUCTION

New ideas and invention are the parts of an Engineer's life. Engineers always try to look upon the problem from a technical point-of-view.

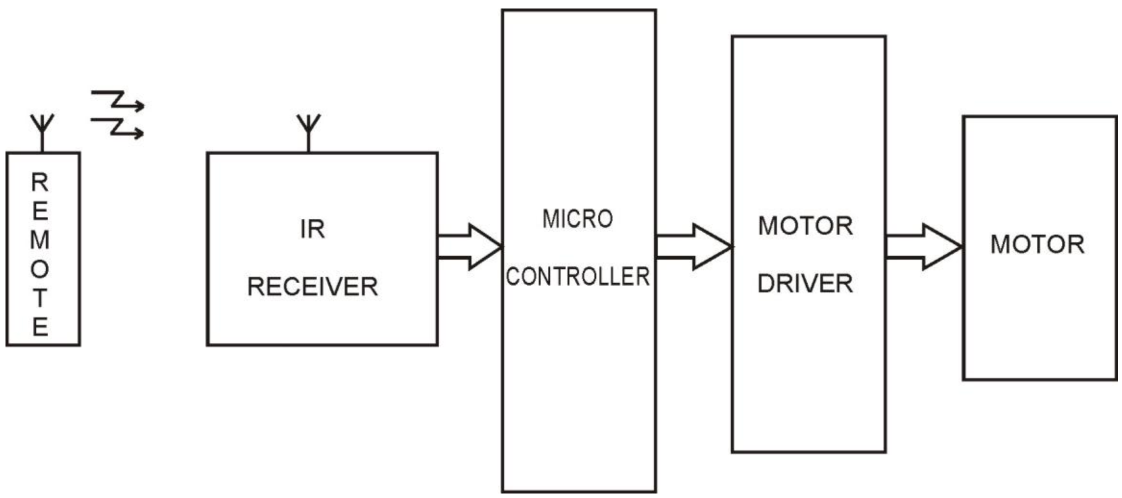
An engineering project is a perfect balance cocktail of the practical aspects of the humanities and economics. The project provides an engineer a deal with a problem existing in society.

Security is prime concern in our day-days life. Every one wants to be as much as secure as to be possible. An access control system forms a vital link in a security chain.

The micro controller based automatic gate lock presented here is an access control system that allows only authorized person enters predetermined number via the keypad, the gate can be open / closed at the end of preset delay. The relay re-energizes and the door gets locked again. We can open as closed the gate as per whenever we required using remote. Using ‘9’ no. key we can open the gate and by using ‘1’ no. key we can close the gate.

If the entered password is correct the unit gives a longer beep of one second. And if the entered password is wrong it gives three small beeps. When the code has been incorrectly entered in a row, the code lock will switch to alarm mode. In this state, all key pressed are ignored for one minute and at the same time the buzzer will sound the alarm signal. This functional any attempt by ‘hackers’ to quickly the try a large number of codes in a sequence. The secret code can be changed by time after entering the current code (Master code). We can reset all the system by using ‘8’ no. key.

BLOCK DIAGRAM



Block diagram

COMPONENT LIST

1) Micro controller - AT 89C2051

2) L298

3) Motor – 10 rpm

4) Buzzer

5) IR reciver

6) Remote control (90 channel)

7) Resister Network (10 ku)

8) Capacitor

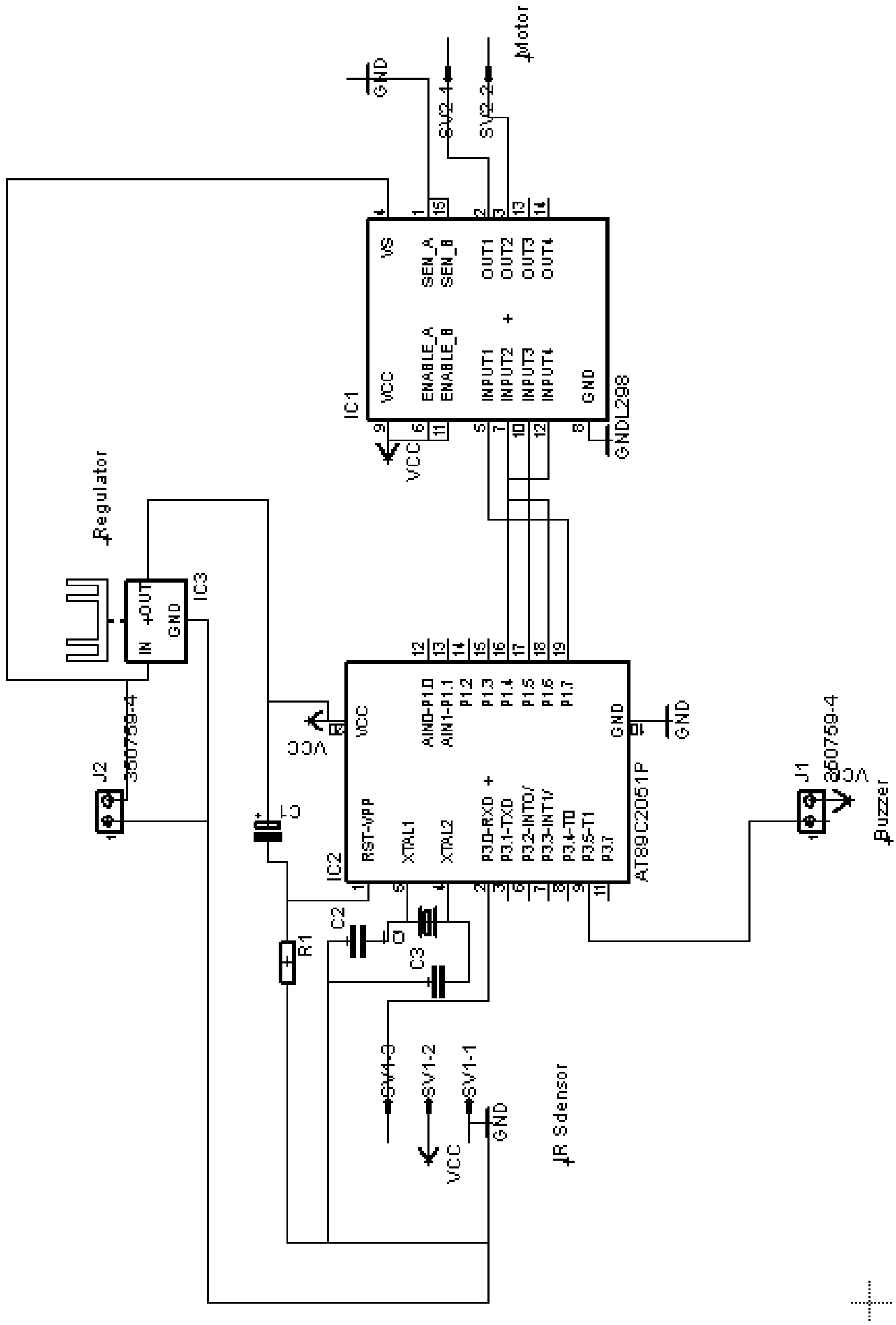
9) Resistor

10) Power supply (tranaformer 12-0-12)

11) Diode 1N4007

12) IC 7805

CIRCUIT DIAGRAM & WORKING



WORKING

Figure shows circuit diagram of gate control system using remote. The main controlling unit is microcontroller AT89C2051. This chip is having tow port P1 & P3. P1 is 8 bit port and P3 is 7 bit port. Port 1 is connected to driver IC L298. Port P3.1 is connected to IR receiver and port P 3.5 is connected to buzzer.

Gate controlling motor is connected to driver IC L298. The driver IC is used to amplifier the current required to drive the motor. The direction controlling command is given by microcontroller on the basis of signal received from IR sensor. The signal receive the IR sensor is the command send by IR remote. This command is denoted by micro controller and action is taken so that gate should open or closed as per the command.

For the controller action the programmed is written in assembly language and burn to the chip this is main controlling programmed for whole operation of the system.

IC PROGRAMMING

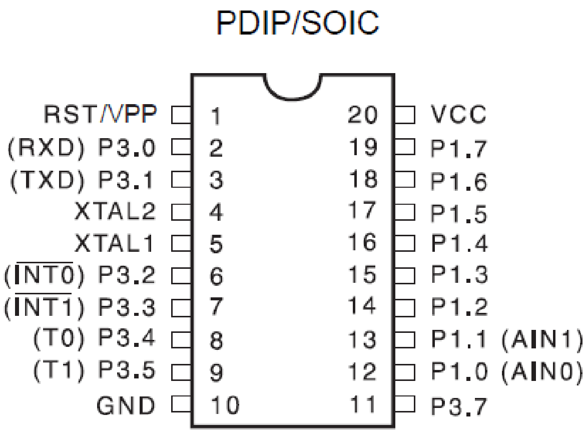
```
;PROGRAM TO SEND COMMAND ON PORT P0


INCLUDE 89c52.mc



COUNT     equ 11H              ;Count
ADDR      equ 12H              ;Device address
CMD equ 13H           ;Command
IR    EQU P3.0        ;IR Receiver connected to this pin
led EQU P1.3
IN1 EQU P1.7
IN2 EQU P1.6
IN3 EQU P1.5
IN4 EQU P1.4
buz EQU P1.0
      org 00H                ;Start of prog
      ;mov SWport,#00H ;switch all relays off!
      mov sp,#50H            ;Stack pointer initialization


main:
  ;MOV p1,#00h
  CLR IN1
  CLR IN2
  CLR IN3
  CLR IN4
  CLR LED
  MOV R0,#00H
  MOV R1,#00H
  INIT:
      jb IR,$                ;Wait for first bit
      mov VAR1,#255          ;3.024mS delay
      djnz VAR1,$
      mov VAR1,#255
      djnz VAR1,$
      mov VAR1,#255
      djnz VAR1,$
      mov VAR1,#255
```

```
        djnz VAR1,$
        mov VAR1,#255
        djnz VAR1,$
        mov VAR1,#100
        djnz VAR1,$
        mov c,IR                ;Read Flip bit
        mov FLIP,c
        clr A
        mov COUNT,#5            ;Count for address
fadd:
        mov VAR1,#255           ;1.728mS delay for each bit
        djnz VAR1,$
        mov VAR1,#255
        djnz VAR1,$
        mov VAR1,#255
        djnz VAR1,$
        mov VAR1,#4
        djnz VAR1,$
        mov c,IR
        rlc a
        djnz COUNT,fadd
        MOV ADDR,A              ;Save the address
        clr a
        mov COUNT,#6            ;Count for Command
fcmd:
        mov VAR1,#255           ;1.728mS Delay for each bit
        djnz VAR1,$
        mov VAR1,#255
        djnz VAR1,$
        mov VAR1,#255
        djnz VAR1,$
        mov VAR1,#4
        djnz VAR1,$
        mov c,IR
        rlc a
        djnz COUNT,fcmd
        mov TEMP,CMD            ;Save the old command
        mov CMD,a               ;Save the new command
        ;MOV p1,addr
        ;ACALL delay

        ;MOV p0,cmd
```

```
CJNE A,#04H,PAS1
MOV R1,#04H
SETB LED
LCALL DELAY
CLR LED
LJMP INIT
PAS1:
CJNE R1,#04,PAS2
CJNE A,#05H,PAS2
MOV R1,#05H
SETB LED
LCALL DELAY
CLR LED
LCALL DELAY
SETB LED
LCALL DELAY
CLR LED
SKIP:
LJMP INIT
PAS2:
CJNE R1,#05H,PAS3
CJNE A,#06h,PAS3
MOV R1,#06H
SETB LED
LCALL DELAY
CLR LED
LCALL DELAY
SETB LED
LCALL DELAY
CLR LED
LCALL DELAY
SETB LED
CLR LED
LCALL DELAY
LJMP INIT
PAS3:
CJNE R1,#06H,PAS5
CJNE A,#08H,PAS6
LJMP MAIN
PAS6:
CJNE A,#01H,PAS4
SETB IN1
CLR IN2
```

```
LCALL DELAY
CLR IN1
LJMP INIT
PAS4:
CJNE A,#09H,PAS5
CLR IN1
SETB IN2
LCALL DELAY
CLR IN2
LJMP INIT

PAS5:
INC R0
CJNE R0,#05H,PAS7
MOV 30H,#20H
BUZ1:
CLR BUZ
LCALL DELAY
SETB BUZ
LCALL DELAY
DJNZ 30H,BUZ1
LJMP MAIN

PAS7:
LJMP INIT


;LCALL delay

DELAY:
MOV R5,#FFH
HERE:
MOV R6,#FFH
DJNZ R6,$
DJNZ R5,HERE
RET


END                ;End of program
```

PIN CONFIGURATION



Pin Description

VCC

Supply voltage.

GND

Ground.

Port 1

Port 1 is an 8-bit bidirectional I/O port. Port pins P1.2 to P1.7 provide internal pullups. P1.0 and P1.1 require external pullups. P1.0 and P1.1 also serve as the positive input (AIN0) and the negative input (AIN1), respectively, of the on-chip precision analog comparator. The Port 1 output buffers can sink 20 mA and can drive LED displays directly.

When 1s are written to Port 1 pins, they can be used as inputs. When pins P1.2 to P1.7 are used as inputs and are externally pulled low, they will source current (IIL) because of the internal pullups. Port 1 also receives code data during Flash programming and verification.

Port 3

Port 3 pins P3.0 to P3.5, P3.7 are seven bidirectional I/O pins with internal pullups. P3.6 is hard-wired as an input to the output of the on-chip comparator and is not accessible as a general purpose I/O pin. The Port 3 output buffers can sink 20 mA. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL) because of the pullups. Port 3 also serves the functions of various special features of the AT89C2051 as listed below:

Port Pin Alternate Functions

P3.0 RXD (serial input port)

P3.1 TXD (serial output port)

P3.2 INT0 (external interrupt 0)

P3.3 INT1 (external interrupt 1)

P3.4 T0 (timer 0 external input)

P3.5 T1 (timer 1 external input)

Port 3 also receives some control signals for Flash programming and verification.

RST

Reset input. All I/O pins are reset to 1s as soon as RST goes high. Holding the RST pin high for two machine cycles while the oscillator is running resets the device. Each machine cycle takes 12 oscillator or clock cycles.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

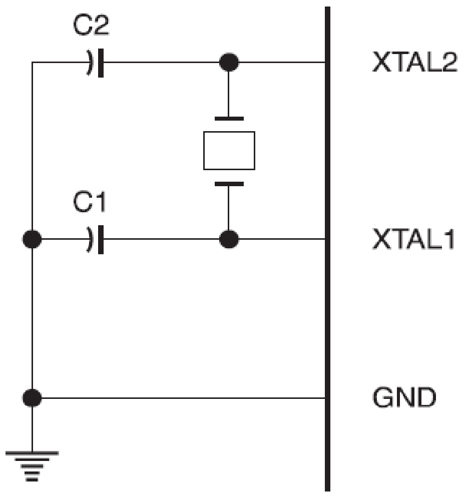
Output from the inverting oscillator amplifier.

Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2.

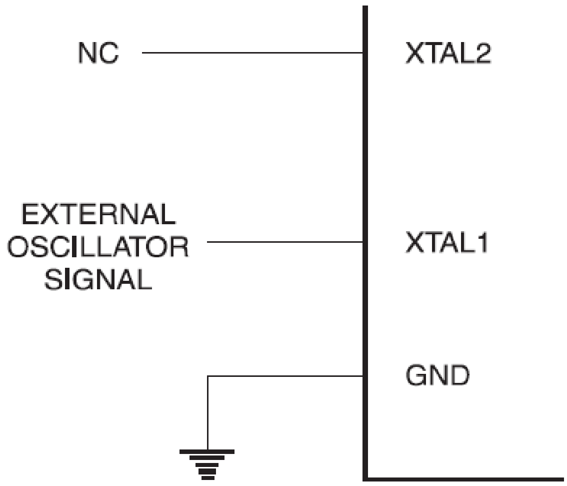
There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Figure 1. Oscillator Connections



Note: C1, C2 = 30 pF □ □10 pF for Crystals = 40 pF □ □10 pF for Ceramic Resonators

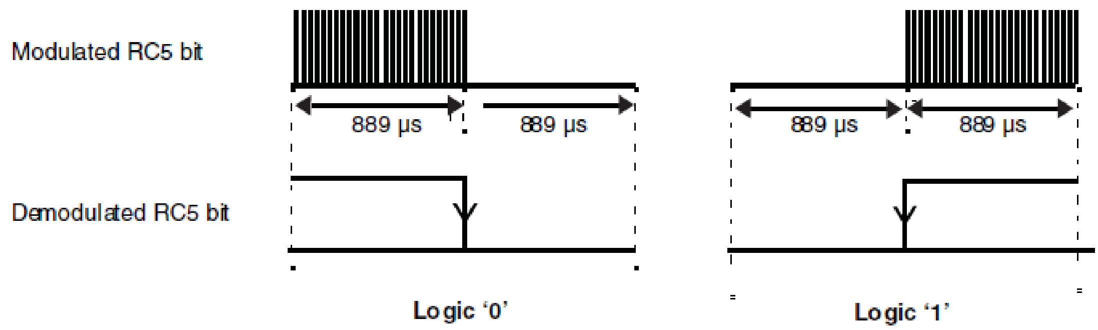
Figure 2. External Clock Drive Configuration



RC5 PROTOCOL -

The RC5 code is a 14-bit word, it uses bi-phase modulation (also called Manchester coding) of a 36 kHz IR carrier frequency. All bits have an equal length of 1.778 ms, with half of the bit time filled with a burst of the 36 kHz carrier and the other half being idle. A logical zero is represented by a burst in the first half of the bit time. A logical one is represented by a burst in the second half of the bit time (refer to *Figure 1.*). The duty cycle of the 36 kHz carrier frequency is 33% or 25% which reduces power consumption.

Figure 1. RC5 bit representation



The RC5 frame can generate 2048 (32x64) different commands organized on 32 groups, each group has 64 different commands. A RC5 frame contains the following fields (An example of RC5 frame is shown in *Figure 2.*):

● **Start bit (S):** 1 bit length, always logic 1.

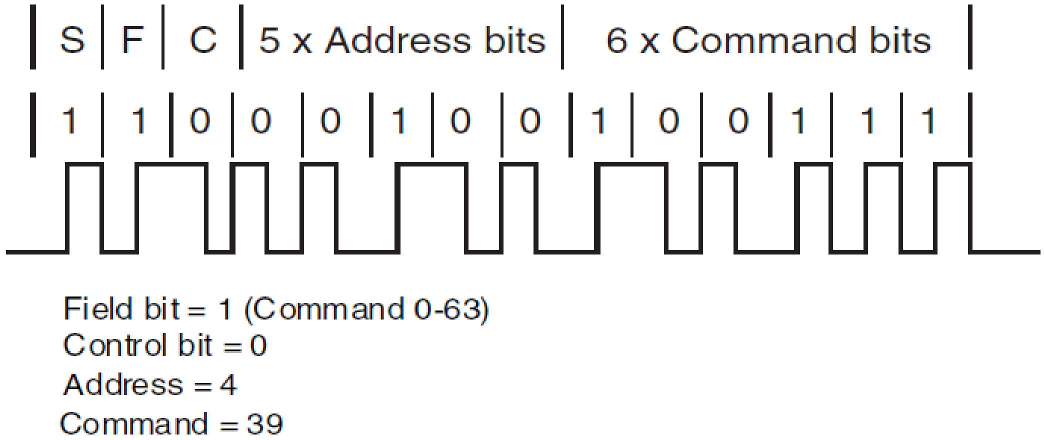
● **Field bit (F):** 1 bit length, which denotes whether the command sent is in the lower field (logic 1 = 0 to 63 decimal) or the upper field (logic 0 = 64 to 127 decimal). The field bit was added later by Philips when it was realized that 64 commands per device were insufficient. Previously, the field bit was combined with the start bit. Many devices still use this original system.

● **Control bit or Toggle bit (C):** 1 bit length, which toggles each time a button is pressed. This allows the receiving device to distinguish between two successive button presses (such as "1", "1" for "11").

● **Address:** 5 bits length, that selects one of 32 possible systems.

● **Command:** 6 bits length, that (in conjunction with the field bit) represents one of the 128 possible RC-5 commands.

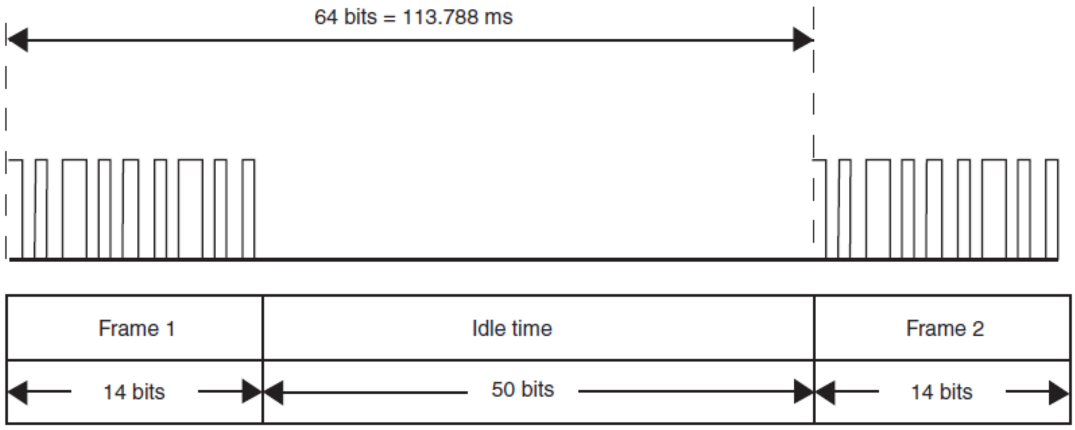
Figure 2. Example of an RC5 frame



To avoid frame collisions, an idle time is inserted between two frames with a specific width (see *Figure 3.*).

The idle time is defined as 50 bits wide. So the periodicity of a frame is 64 x 1 bit width: 64 x 1.778 = 113.792 ms (exactly 113.788 ms).

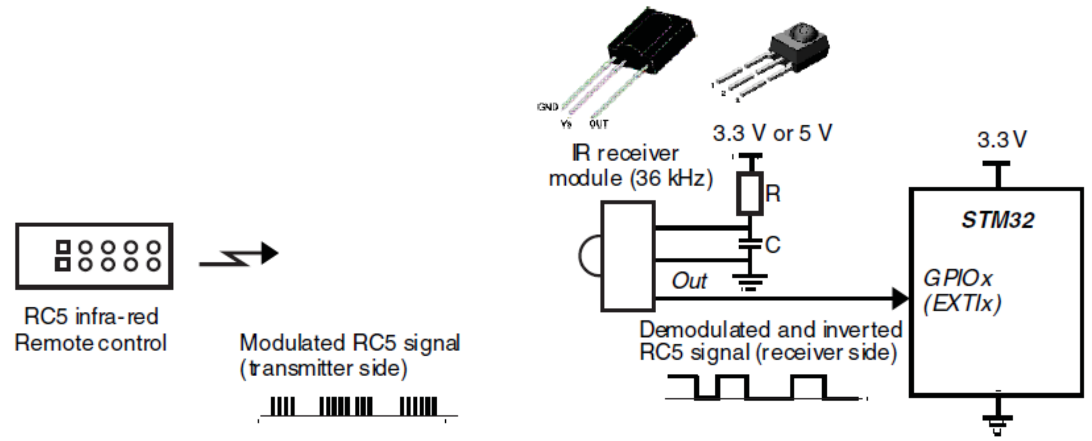
Figure 3. RC5 idle time



Hardware considerations

To improve noise rejection, the pulses are modulated at 36 kHz. The easiest way to receive these pulses is to use an integrated IR-receiver/demodulator module like the TSOP1736 (5 V supply version) or TSOP34836 (3.3 V supply version) or other equivalent part number (refer to *Figure 4.*). These are 3-pin devices that receive the infrared burst and output the demodulated bit stream on the output pin which is connected directly to one of the STM32 microcontroller's GPIO pins. The GPIO pin used is selected by the user (refer to the section *Section 3.2.3 on page 8*). If TSOP1736 is used, the selected GPIO should be Five volt Tolerant (FT). The output of the IR module is inverted compared to the transmitted data (the data is idle high and logic '0' becomes logic '1' and vice versa)

Figure 4. Hardware configuration

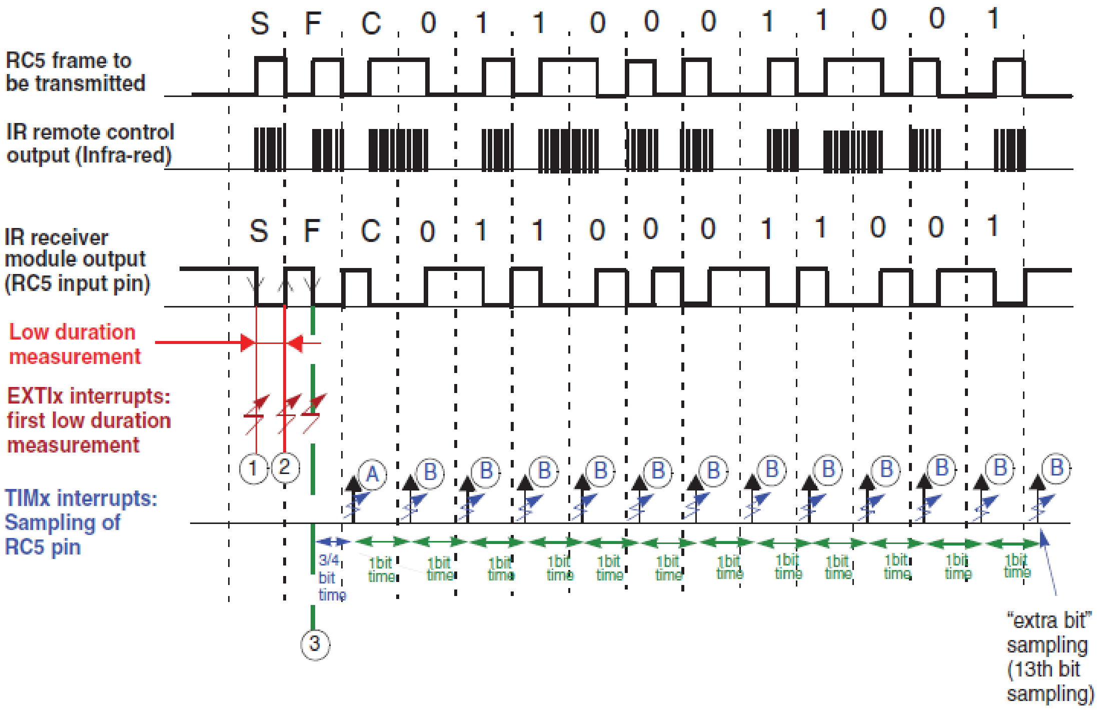


Software implementation

RC5 frame reading mechanism

Figure 5. shows how the RC5 frame is received. Principally, two of the STM32 microcontroller's embedded peripherals are used for this purpose: EXTI and a timer (TIMx). The STM32 pin connected to the IR module's output pin can be any GPIO selected by the user (see *Section 3.2: How to use the RC5 library on page 7*).

Figure 5. RC5 frame reception mechanism



EXTI interrupt events

The EXTI interrupt is used to start and stop TIMx in order to measure the first low duration to validate the header timing of the RC5 frame. Refer to events 1, 2 and 3 in *Section Figure 5.: RC5 frame reception mechanism on page 6*

- First EXTI interrupt event (1): the TIMx counter is initialized and enabled.
- Second EXTI interrupt event (2): the TIMx counter is disabled, read and then initialized.

The value read from the counter gives the measured duration. The 3rd execution of the
EXTI interrupt depends on the measured duration:

– If the duration is within the tolerance range of one half bit time, the EXTI is not disabled and the EXTI interrupt occurs for the 3rd time, which enables TIMx. TIMx then starts sampling the RC5 data. In this case the *Field* bit will be recognized as a logical 1.

– If the duration is within the tolerance range of one bit time, the EXTI is then disabled at this moment and the TIMx update event interrupt as well as the TIMx counter are enabled to start sampling the RC5 data. In this case the *Field* bit willrecognized as a logical 0. If the duration seems to be a glitch, the system will be initialized for the next RC5 frame.

- Third EXTI interrupt event (3): this interrupt occurrence depends on the duration of the first low duration, see (2). When the interrupt occurs, TIMx is enabled and starts sampling the RC5 data.

TIMx interrupt events

TIMx is used to sample each bit of the RC5 frame after checking the timing of the first low duration of the frame. TIMx interrupts are executed 13 times during an RC5 frame in order to sample all the bits.
The Start bit (S) and Field bit (F) are not sampled by TIMx and an « extra bit » is sampled at the end of the RC5 frame to be sure that all bits have been received and an idle state is present.

- TIMx interrupt event (A): at this time, the RC5 pin is sampled by a single reading of the GPIO input data register. In this interrupt service routine, the TIMx is configured to generate a periodic interrupt each bit time.
- TIMx interrupts event (B): at this time, the RC5 pin is sampled by a single reading of the GPIO input data register and the interrupt service

routine checks if the number of data bits has reached 13 (n = 13: 14-2+1) If yes, the TIMx counter and the TIMx update interrupt are disabled.

As we can see, reading from the GPIO input data register directly reflects the value of the bit. If the read value is at low level this implies that the bit value is logic ‘0’. If the value read is at high level this implies that the bit value is logic‘1’.

How to use the RC5 library

The RC5 driver is very simple to use. There are four functions accessible for the user.

RC5_Receiver_Init() function

This function is intended to initialize the different peripherals used by the RC5 driver: GPIOs, EXTI and TIMx. It should be called after the user clock configuration.

RC5_Sample_Data() function

This function is used to sample the RC5 data. It should be called in the RC5_TIM_IRQ_Handler routine (TIMx_IRQHandler) in the stm32f10x_it.c file. The RC5_IR_Receiver.h file should be included in the stm32f10x_it.c file. By default, TIM2 is used. You can use any timer (TIMx in the list below) by modifying the defines in the RC5_IR_Receiver.hfile(path:
\STM32F10x_AN3174_FW_VX.Y.Z\Libraries\
STM32F10x_RC5_Emul_Receiver_Lib\inc\RC5_IR_Receiver.h)

RC5_MeasureFirstLowDuration() function

This function measures and validates the first low duration of the RC5 frame. When this timing is in the range of the allowed timings, the function enables the RC5 frame sampling. This function should be called in the appropriate EXTI interrupt handler (in stm32f10x_it.c file) depending on the GPIO used for the RC5 input pin. By default, GPIOB.01 is used as the RC5 input pin. You can use any GPIO by modifying the defines in the RC5_IR_Receiver.h file path:

\STM32F10x_AN3174_FW_VX.Y.Z\Libraries
\STM32F10x_RC5_Emul_Receiver_Lib\inc\RC5_IR_Receiver.h).

RC5_Decode() function

This function is intended to be called in the user application. It decodes the RC5 received messages. It returns a structure that contains the different values of the RC5 frame.

```
typedef struct
{
__IO uint8_t ToggleBit;  /*   Toggle bit field */
__IO uint8_t Address;    /*   Address field */
__IO uint8_t Command;    /*   Command field */
} RC5Frame_TypeDef;
```

RC5_decode() should be called when the RC5_FrameReceived flag is equal to YES.

Example of usage:

```
/* System Clocks Configuration */
RCC_Configuration();
/* Initialize RC5 reception */
#define RC5_TIM_IRQ_Handler TIM3_IRQHandler
```

● You can choose any of the STM32F10x family timers.

RC5_MeasureFirstLowDuration() function

This function measures and validates the first low duration of the RC5 frame. When this timing is in the range of the allowed timings, the function enables the RC5 frame sampling. This function should be called in the appropriate EXTI interrupt handler (in stm32f10x_it.c file)

depending on the GPIO used for the RC5 input pin. By default, GPIOB.01 is used as the RC5 input pin. You can use any GPIO by modifying the defines in the RC5_IR_Receiver.h file path: \STM32F10x_AN3174_FW_VX.Y.Z\Libraries\
STM32F10x_RC5_Emul_Receiver_Lib\inc\RC5_IR_Receiver.h).

RC5_Decode() function

This function is intended to be called in the user application. It decodes the RC5 received messages. It returns a structure that contains the different values of the RC5 frame.

```
typedef struct
{
__IO uint8_t ToggleBit; /* Toggle bit field */
__IO uint8_t Address; /* Address field */
__IO uint8_t Command; /* Command field */
} RC5Frame_TypeDef;
```

RC5_decode() should be called when the RC5_FrameReceived flag is equal to YES.

Example of usage:

```
/* System Clocks Configuration */
RCC_Configuration();
/* Initialize RC5 reception */
```

PCB DESIGN

INTRODUCTION

Printed circuit board is a piece of art. The performance of an electronic circuit depends upon the layout and design of PCB. The PCB design of the circuit operation should be very precise to work it properly. The soldered point should be small enough so that any stray between these points should not exist. Also high package density of components can produce stray which should be avoided by proper circuit designing and components should be spread in such a way that two-component produce minimum stray. Also the track of the PCB, soldering points and components mounting should be very correct and that will be of great help to success the project.

Making such precise PCB is easy. For preparing the PCB layout, we used the PCB layout manufacturing by the Vega company with a help of computerized equipment. We can not use readymade PCB for our project. The trackside of the PCB is shown in figure.

To make the PCB with professional touch, the general method that should be carried out is as follows.

LAYOUT PLANNING:

The layout of the PCB has to incorporate all information on the board, before one can proceed further for the artwork preparation. This planning procedure depends on many factors.

LAYOUT SCALE:

Depending upon the accuracy required artwork produced should be at 1:1 or 2:1 scale. Accordingly the size of the artwork will be equal to four times or sixteen times of that actual PCB. The layout is best prepared on the same scale as artwork.

LAYOUT SKETCH:

The end produced of the layout design is the pencil sketched component and conductor driving, which is called layout sketch. It contains all relevant information for preparation of artwork.

Besides the components outlines, components holes and interconnection line (patterns) the layout should also include the information on.

- Diameter of component hole, IC transistor pads.
- Minimum spacing between the conduction lines that must produced.
- Standard conductor widths all should be used for specified application.

ARTWORK:

Preparation of artwork is considered as first step in preparation of PCB. Following steps are included while designing the artwork. A polyester foil and tracing paper may be used. Basic methods of preparing artwork are:

1. Ink the drawing.
2. Using block tapes and sticking patterns.
3. Using red and blue transparent tapes.

The artwork is then converted to photonegative of proper size.

PHOTOGRAPHIC FILM PRODUCTION:

A film negative of actual PCB is required. The following equipments are required for photo reduction.

- Camera
- Lens system with minimum distortion.
- Light arrangement at the back of the negative to provide contrast of 1000:1.
- From light arrangement with a contrast of 10:1.

The range of reduction of artwork is 2:1 to 4:1. A special darkroom is required to prepare negative. Room temperature should be less than 25°C, sufficient circulation of dust free air. The room should be light proof.

PREPARATION OF SINGLE SIDE PCB:

In single side PCB, the conductor tracks run only on one side of copper cladded board. Thus crossing of conductor is not allowed. Figure shows the steps to be followed in preparing single sided PCB.

PROCEDURE:

Base material is selected and it is mechanically and chemically cleaned. Then the photo resists solution, which when exposed to light of proper wavelength, changes their solubility in the developers, and is uniformly applied. There are two types of photo resists:

1. Negative acting.
2. Positive acting.

Coating of photo resists is done by:

1. A spray coating.
2. Dip coating.
3. Roller coating.

The coated paper clad laminate and film negative are kept in glass frame in intimate contact with each other. The assembly is exposed to ultraviolet light for three minutes. The exposed board is rinsed in the developer tank. Proper developer has to be used for particular photo resists. Then the PCB is dyed in a tray, the dye reveals the faults due to contrast, which are then removed by retouching.

Etching removes the unwanted copper. The spray etching gives the best results. Cupric chloride is regenerative and is thus preferred to ferric chloride. The etching solution is kept agitating by circulating low pressure air at the bottom of the tank through air nozzles. Then the board is drilled with a high speed drilling machine. Centering of hole on the land pad is done automatically.

SOLDERING AND SOLDERING TECHNIQUE

There are basically two types of soldering techniques:

- Manual soldering with iron.
- Mass soldering.

SOLDERING WITH IRON:

The surface to be soldered must be cleaned and fluxed. The soldering iron is switched on and allowed to attain soldering temperature. The solder in the form of wire is applied near the component to be soldered and heated with iron. The surfaces to be soldered are filled, iron is removed and the joint is cooled without disturbing.

The following precaution should be taken while soldering:

✓ Use always an iron plated copper core for soldering iron.

- ✓ Clean the component leads and copper pads before soldering.
- ✓ Apply solder between components leads, PCB pattern and tip of soldering iron.
- ✓ Use optimum quantity of solder so that θ is small.
- ✓ Remove flux residues from PCB with solvents like isopropyl alcohol.
- ✓ Use Sn 60 (Sn 60%, Pb 40%) or Sn 63 (Sn 63%, Pb 37%) composition solders.

SOLDERING:

Soldering is a process used for jointing metal parts. It is necessary to use molten metal known as solder.

During soldering, relative positioning of the surfaces to be joined, wetting of this surface with molten solder and cooling time for solidification is important.

MASS SOLDERING:

Mass soldering is used where large numbers of joint are to be soldered simultaneously. It has high productivity, and reliability of final assemblies.

DIP SOLDERING:

A prefluxed PCB with assembled components is dipped vertically into clean solder bath to depth. It uses 60% tin and 40% lead. The board is kept in bath for 2-3 seconds and angled path should be followed while taking out it.

WAVE SOLDERING:

The assembled board is applied flux. It is preheated while passing through conveyor belt to 110°C and then to 140°C. It then passes over a λ wave of solder.

Thus the various steps in the mass soldering are as follows.

1. Assembling the board.
2. Flux application.
3. Flux drying or preheating.
4. Cooling
5. Flux removal.

DRAWING PREPARATION

With the circuit diagram & component is hand, draw the complete layout on a plane sheet of tracing paper in the same way as if you are assembling the circuit keep the lives one side, and line on the other side as bar as possible. When all the components are mounted on the tracing paper take a etch pen to mark the connection on the tracing paper.

Remove the components and give finishing touches to the marking so that all connections are shown with equal width. The layout of the figure is shown in figure.

PRINTING OF PCB

The drawing so prepared has to imposed over the glass epoxy. Take a PCB terminated sheet and cut the of required size of PCB by using hacksaw place the glass epoxy plate sheet on a table, keeping the glass epoxy side on rub away the dirt, grease and oxide wish a sand paper. Now keep carban paper of the same size on PCB taking glass epoxy surface on the top carban paper. Since the tracing paper is transparent you can now reproduce carbon print over the PCB. After tracing the PCB layout now paint the tracks wish the help of oil paint and brush, keep plate in open to dry. After the paint on a copper side has dried, check the drawing carefully, excess paint should be scratched off wish of a blade.

ETCHING OF PCB

In a tray, take water and mix a few tea spoons of ferric chloride powder and few drop of HCL. Immerse/dip the PCB in this solution keep the PCB in this solution for about 40-50 min.

Reaction - 2FeC13+2Cu=2CuC12+Fe2Cl

Observe the changing color of copper surface. Take out the PCB from the solution only when the unmarked portion of copper is completely dissolved in this solution wash the PCB wish water. After washing PCB, remove the paint with a soft piece of cloth or cotton. Now the plate is what we call it as printed circuit board.

FLUX

It is used to dissolve & remove the oxide & other contaminators frem the surface of PCB, & to remove the ability of solder to spread over evenlynon the surface of PCB. After soldering the coating of flux is mixed is removed and fabrication is completed.

DRILLING, MOUNTING AND SOLDERING.

After the etching process drilling is done for mounting the components. Drill the board by using hand drill or machine drill. Before inserting the leads of the components are placed on the irrespective position (according to the circuit Diagram) this process is called as component mounting.

Now the next process is soldering. In this process, the leads of components are joined/ soldered with the copper tracks of PCB. For this tussible alloy metal which is known as 'SOLDERING WIRE[1] is required soft solder has 37% of lead and 63 % of zinc and is used because of its excellent drying action. Its melting point is very low. It gives mechanically strong point for soldering the components, soldering gun is used. Flux is used as an inorganic solvent.

ADVANTAGES

1. Fully Secured.
2. Sounds the buzzer if incorrect password is entered.
3. Informs the securities by alarming if incorrect password is entered more than four times.
4. Works off AC mains or batteries.
5. Fine design.

FEATURES

- Operating voltage -12V AC / DC (nominal)
- Operating current - 100mA with no relays operated / 500mA with relays operated (Aprox)
- Access Time (Gate opening time) - 10 Second Or Latch (Continuous)
- 3 x 4 Keypad Type - Remote type
- 3 Digits Secret Code - (0 To 9)
- On board serial EEPROM (no require battery back up)
- Buzzer to annunciate a key press, error situation and Alarm

CONCLUSION

Security access control (SAC) is an important aspect of any system. Security access control is the act of ensuring that an authenticated user accesses only what they are authorized to and no more. The bad news is that security is rarely at the top of people's lists, although mention terms such as data confidentiality, sensitivity, and ownership and they quickly become interested. The good news is that there is a wide range of techniques that you can apply to help secure access to your system.

This project deals with the security access system. A automatic lock is prepared here. Every time to open the gate, correct password is needed. Input of wrong password will not allow the user to enter and inform the surrounders by sounding alarm.

As a result it is found that the system performed well in stopping unauthourized access.

REFERENCES

1. “Access Control System” - Kevin D. Mitnick
2. “Security Engineering” – Ross Andorson, Willey Publication
3. Gate Access Control System – Product information by Raviraj Technologies, Mumbai
4. www.google.com
5. www.wikipedia.org
6. www.datasheet4u.com
7. www.electronicsforyou.com

CONTENTS