

AN-Najah National University
Faculty of Engineering
Electrical Engineering Department



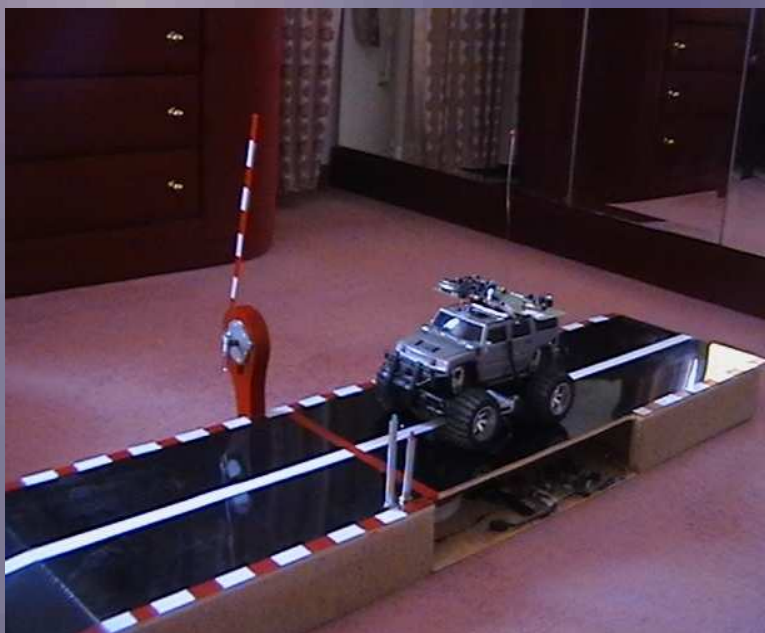
Inductive Link

Name : Asad Faraj El-Bizreh

Belal Bani Jaber

Mothana Hakawati

Dr. Mazen Rasekh



2008-2009



Contents

Chapter one:

Introduction:.....

**General Introduction*

**Project Introduction*

Chapter Two:

Solutions:

**Possible Solutions*

**Solution Evaluation*

**Communications Link Range*

**Ring Frequency*

Chapter Three:

Steps of the project:.....

1) Microprocessor of Transmitter (Arduino)

2) Power amplifier (common emitter)

3) Ring generator (transmitter) :

** Transmitter Network*

** Network Driver*

** Transmitter Coil Design*

4) the coils :

** Mutual inductance*

** Data Encoding Scheme*

5) Ring receiver

6) Schmitt triggers (pulse generators)

7) 555 Monostable

8) Microprocessor of Receiver (Arduino)

9) Stepper Motor (to open the door of the car park)

** Summary in pictures*



Chapter one

Introduction

**General Introduction:*

How do I send sensor information from one side of a structure to the other side, without penetrating the walls? Perhaps the structure is the concrete foundation of a home or a high rise building. It might also be a standard wood frame building. The data I imagine sending would not require a high data rate. Perhaps the data would be weather information such as: temperature, humidity, wind speed and barometric pressure. Or maybe it is sensor information for a lawn irrigation system or temperature data for a roof mounted solar heating system. The technique might also be used to send useful information such as gas, electricity or water consumption from one side of a building to the other.

Rough design requirements The signal needs to have a range of only three or four feet. The signal needs to be able to penetrate concrete walls that may have steel reinforcing rods. Since it is often difficult to tap into a standard power source outside, the technique needs to be able to operate for long periods of time powered only by a small battery. It may also be possible to extend the useful operating time with the use a small solar array that would keep a battery charged. However, replacing a single battery once each year may be acceptable.

**Project Introduction:*

The project is essentially a practical application on the car park (Garage), which consists of the process of communication between a transmitter located inside the car and a receiver is under the ground (in the entrance of the car park(Garage)) .

As a result of this contact the gate(door) of the car park(Garage) will open automatically.

where the transmitter is inside the car and each car has a specific code and this code will be sent in a continuous from a transmitter system, and when the car come to the entrance of the car park(Garage) the transmitter sends a code of the car by a magnetic field to the receiver located under the ground, as the receiver receives a code of the car and check if this code was one of the existing codes which are allowed to enter the car park, if the car was carrying the code which is allowed to enter then the receiver give an order to open the gate(door) of the car park to the car, and if the code was not part of this codes (codes which are allowed to enter the gate) then the door remains closed and the car don't enter the car park(Garage).



Chapter Two

Solutions

**Possible Solutions:*

Among the possible methods to transfer the data are: radio transmitter, ultrasound, earth ground signals and magnetic coupled coils.



**Solution Evaluation:*

Radio transmitter methods may require too much power. An ultrasound technique might work for a rigid structure such as concrete but will most likely not work for wood frames walls. Earth ground signal methods may be too unpredictable. The best candidate looks like a magnetic coupled technique. The magnetic method would use two coils to establish a communications link. One coil would form the transmitter and one coil would be used as the receiver. The transmitter would be positioned on the outside of the structure and the receive coil on the inside. The two coils would form a primary and secondary of a transformer

**Communications Link Range?*

The system I imagine would not have to have a range of more than about four feet. Such a range should be enough to penetrate most structures. In cases where the signal only has to penetrate a conventional 2 x 4 wood stud wall, a 12 inch range may work fine. The maximum range will occur when the transmitter coil and the receiver coil are in parallel with each other. But, a system could take advantage of a potentially longer range by allowing for some coil misalignment.

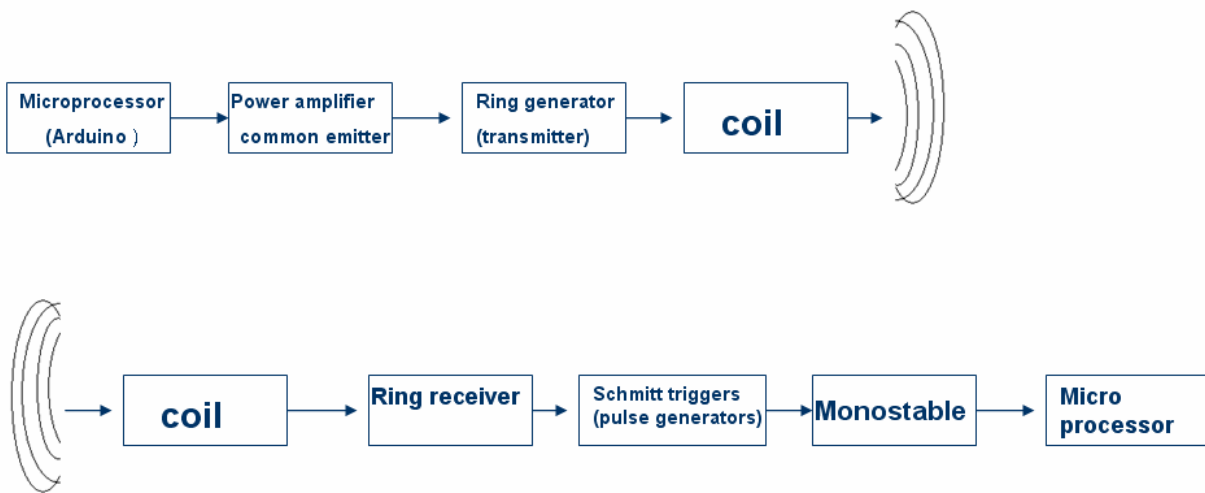
The magnetic field strength of two loosely coupled coils drops off rapidly as the distance between the transmitter and the receive coil is increased. The curve tracks a "cube" function. So, if the distance between the two coils doubles, the signal picked up by the receive coil will be only be one eighth as much.

**Ring Frequency?*

What frequency should I use? Perhaps I could use the US license-free band that exists between 160KHz and 190KHz. Maybe shoot for a 175KHz center frequency. 125KHz is also an international standard for radio frequency identification devices (RFID) so that too might be a candidate. I think I will plan for a 175KHz frequency.

**Steps of the project:*

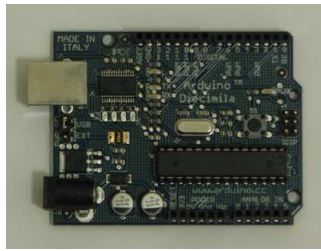
We will follow the next block digrame;



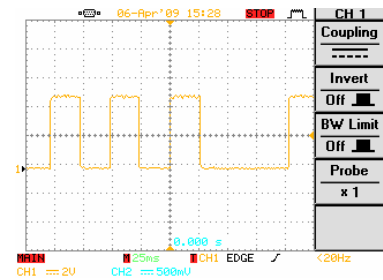
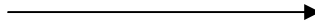
*Steps of the block diagram:

1)Microprocessor of Transmitter (Arduino) :

We will send our code with Microprocessor (Arduino)
By pulses of one's (5 volts) and zero's (0 volt) ,so every car have a code that will continuously send by Arduino to the receiver ,and the output code will be one's and zero's,
And the output of the Arduino is shown:



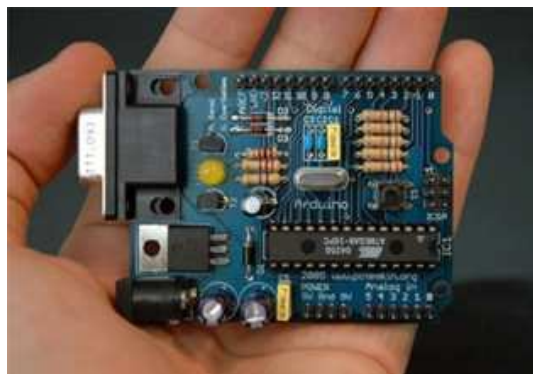
Output code pulses



***We select the Arduino over other Microprocessor because of this properties:**

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It's intended for artists, designers, hobbyists, and anyone interested in creating interactive objects or environments.

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors, and other actuators. The microcontroller on the board is programmed using the Arduino programming language (based on Wiring) and the Arduino development environment (based on Processing). Arduino projects can be stand-alone or they can communicate with software on running on a computer (e.g. Flash, Processing, MaxMSP).



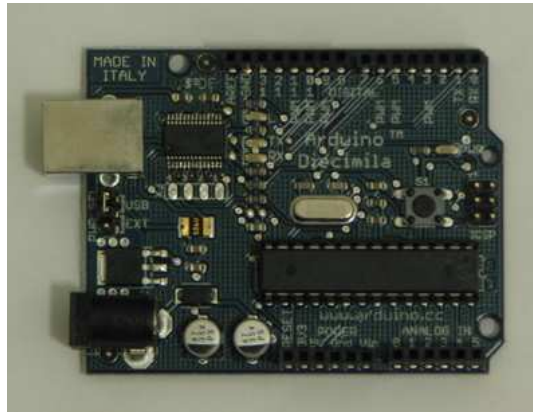
Why Arduino?

Arduino also simplifies the process of working with microcontrollers, but it offers some advantage for teachers, students, and interested amateurs over other systems:

- Inexpensive - Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than \$50
- Cross-platform - The Arduino software runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.
- Simple, clear programming environment - The Arduino programming environment is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well

The beauty of the Arduino is that it uses a very simple yet straight forward and powerful programming language that closely resembles C. Be warned, it does not tolerate sloppy code so you need to keep everything neat and precise. On top of that it is 100% hackable and being open source, you can modify the design, ergo, there are countless Arduino based projects and flavours.

The one I saw was called the Diecimila and had a USB interface, 16KB Flash Memory, 14 digital input/output pins (of which 6 can be used as PWM outputs) and 6 analog inputs.



As you can imagine, this allows you to build a prototype very quickly, and according to my understanding it easily out performs the old 286 computers

2)Power amplifier(common emitter):

After the microprocessor we must amplify the code to nine volts minimum to give the affect on the transmitter circuit and we select the common emitter with two stages to amplify the signal and without invert the original signal

And we select 2N2222 for this amplifier;

***NPN switching transistors 2N2222; 2N2222A**

FEATURES

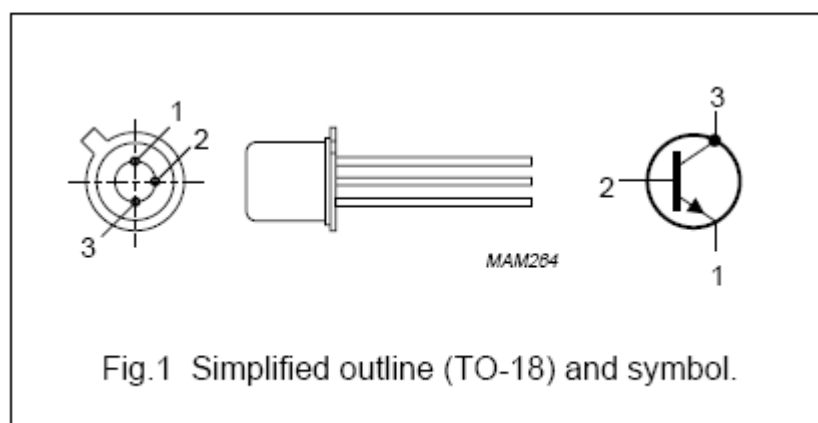
- * High current (max. 800 mA)
- * Low voltage (max. 40 V)

APPLICATIONS

- * Linear amplification and switching.

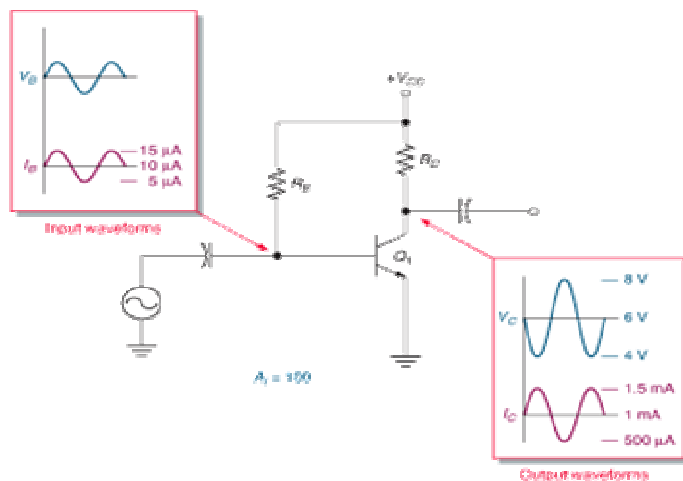
DESCRIPTION

- * NPN switching transistor in a TO-18 metal package.
- * PNP complement: 2N2907A.



*Introduction:

- * Input and output signal voltages are 180° out of phase
- * The input voltage and current are in phase.
- * The input and output currents are in phase. Therefore, output current is in phase with the input voltage.
- * An increase in output current results in a decrease in output voltage, and vice versa (as given by $V_{out} = V_C = V_{CC} - I_C R_C$).

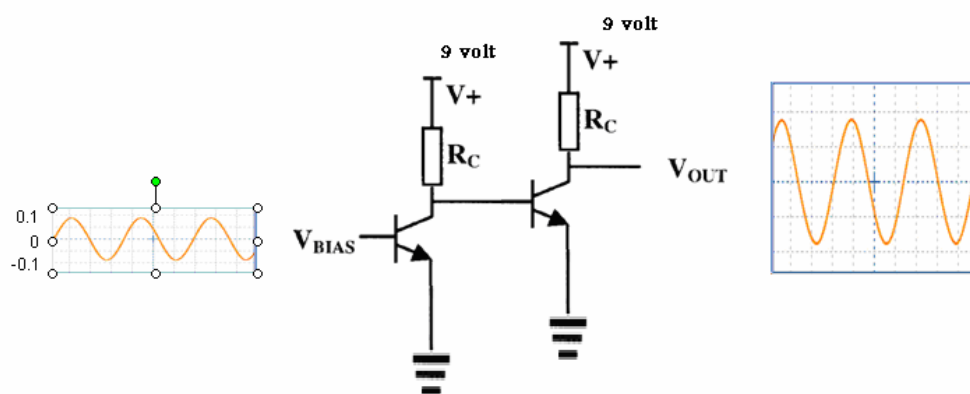


One stage of common emitter

In electronics, a **common-emitter** amplifier is one of three basic single-stage bipolar-junction-transistor (BJT) amplifier topologies, typically used as a voltage amplifier. In this circuit the base terminal of the transistor serves as the input, the collector is the output, and the emitter is *common* to both (for example, it may be tied to ground reference or a power supply rail), hence its name. The analogous field-effect transistor circuit is the common-source amplifier

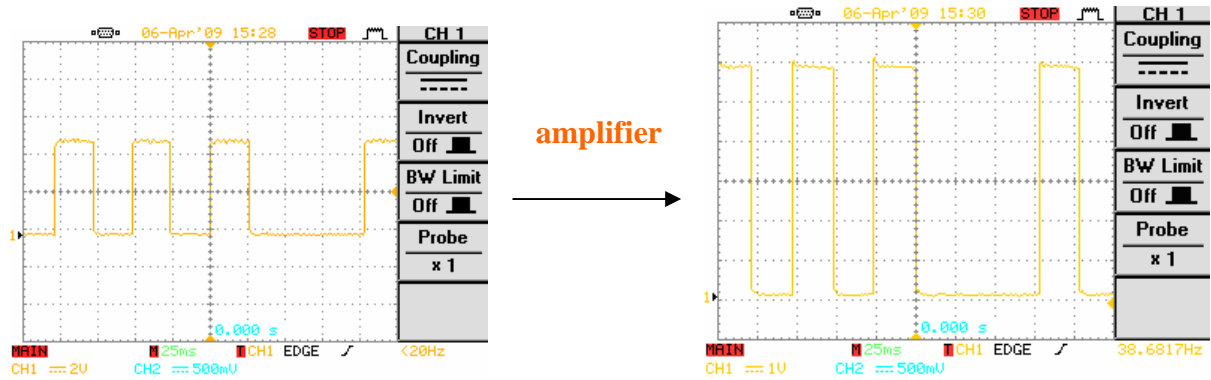
Common-emitter amplifiers can have a very high gain which can vary widely from one transistor to the next. The gain is a strong function of both temperature and bias current, and so the actual gain is somewhat unpredictable. Stability is another problem associated with such high gain circuits due to any unintentional positive feedback that may be present. Other problems associated with the circuit are the low input dynamic range imposed by the small-signal limit; there is high distortion if this limit is exceeded and the transistor ceases to behave like its small-signal model. One common way of alleviating these issues is with the use of negative feedback, which is usually implemented with *emitter degeneration*. Emitter degeneration refers to the addition of a small resistor (or any impedance) between the emitter and the common signal source (e.g., the ground reference or a power supply rail). This impedance R_E reduces the overall transconductance $G_m = g_m$ of the circuit by a factor of $g_m R_E + 1$, which makes the voltage gain

$$A_v \triangleq \frac{v_{\text{out}}}{v_{\text{in}}} = \frac{-g_m R_C}{g_m R_E + 1} \approx -\frac{R_C}{R_E} \quad (\text{where } g_m R_E \gg 1).$$



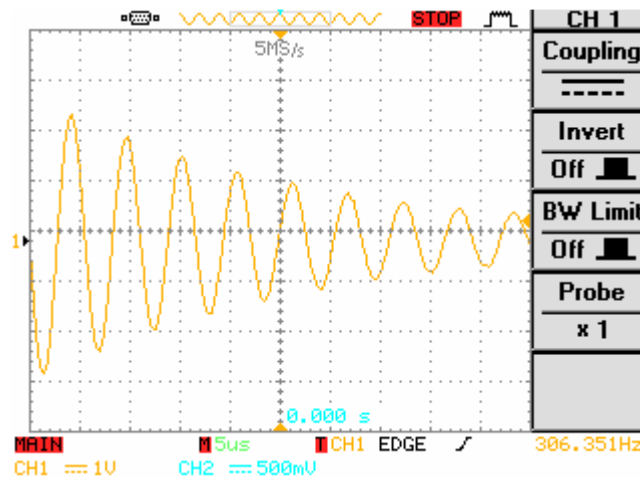
Two stage of common emitter

***and the output of this stage is the code pulse with amplifying as shown:**



3) Ring generator (transmitter):

We will take our code pulse to the transmitter circuit to produce the Rings to transmit our code to the receiver as shown:



Transmitter Network?

The best way to produce a powerful magnetic signal is to drive a series resonant LC circuit with one or more low impedance pulses. The pulses will induce a "ring" frequency in the LC network. The highest magnetic field will occur when the coil current can be maximized. One way to keep coil current high, is to keep the Q of the LC network high. The practical limit of the Q will depend on the source impedance of the circuit driving the LC network. Maybe a Q of about 10 could be the initial goal. With a Q of 10, each pulse will induce a ring signal that will decay to near zero in about 10 cycles.

The transmitter coil needs to be as large as practical, but not too large. Perhaps a coil diameter of about 6 inches would be a good place to start. Maybe I will start out with a 6 inch coil for both the transmitter and the receiver. If that works out, I could then consider reducing the size of one or both of the coils.

And after many experiments we will choose 4 inch coil for the transmitter and 6 inch coil the receiver.

Network Driver?

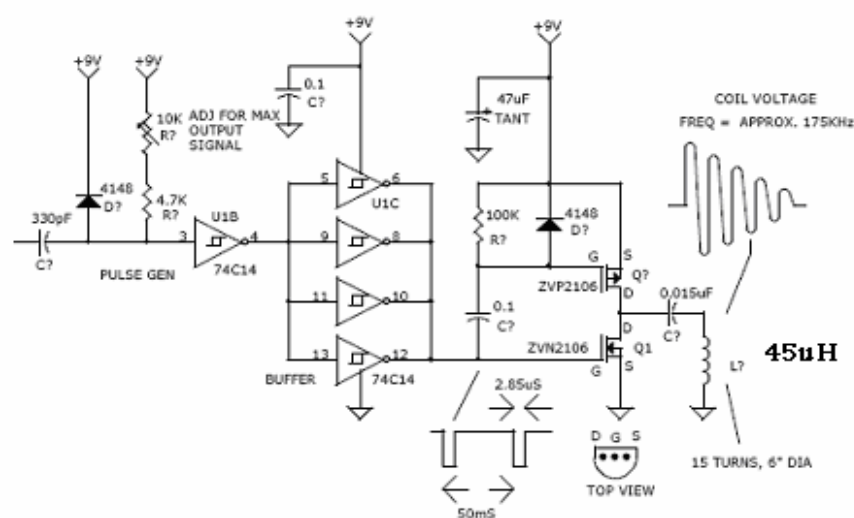
OK, if I use a 175KHz ring frequency, the single drive pulse launched into the LC network will need to be one half the period of 175KHz or 2.86uS. To produce the strongest magnetic field, I need the highest practical coil current and the most number of turns possible. If I use a 9 volt battery and a good pair of small power FETs, then maybe the source current could be about 5 ohms. With a 9 volt battery, that would put the peak coil current at about two amps. If I want a Q of 10, the reactance of the coil at 175KHz will need to be about 10×5 or 50 ohms. Working backwards, if the reactance of the coil is to be 50 ohms at 175KHz, then the inductance of the coil will then need to be about 45uH. To tune the LC network to 175KHz, the series capacitance will then need to be about 0.018uF. To insure I get the best peak current, the capacitor will need a low equivalent series resistance (ESR) rating. A polycarbonate or a mica capacitor should be OK. With a peak current of two amps, the peak to peak voltage across the capacitor would be about $2 \times 2A \times 50$ ohms or about 200 volts. So, the capacitor will need to have a 250 volt rating.

Transmitter Coil Design?

I looked up the formula for calculating the coil inductance of a single layer coil. Based on the equation, I will need about 11 turns of magnet wire for a 4 inch diameter coil to get an inductance of 45uH. So, for my first experiment, I should plan for a 6 inch coil with 11 turns. If necessary I could add or subtract one or two turns to get the circuit to ring at a frequency of 175KHz

The key in the transmitter is the driver circuit.

I think I will try a push-pull type driver that I have used before for driving low impedance loads. The circuit consists of one N-ch FET and one P-ch FET. Both devices are biased in the off state. The two transistors are linked to a signal source that generates a single pulse. During the negative swing of the pulse the P-ch devices applies +9v to the load. When the pulse swings positive, the P-ch devices is turned off and the N-ch device is turned on. The transition time, when one device is turning off and the second is turn on if very fast. So, the circuit is very efficient at delivering current pulses to a low impedance load, such as a series resonant LC circuit



175KHz RING GENERATOR

4) the coils:

I looked up the formula for calculating the coil inductance of a single layer coil. Based on the equation, I will need about 11 turns of magnet wire for a 4 inch diameter coil to get an inductance of 45uH. So, for my first experiment, I should plan for a 6 inch coil with 11 turns. If necessary I could add or subtract one or two turns to get the circuit to ring at a frequency of 175KHz

The transmitter coil needs to be as large as practical, but not too large. Perhaps a coil diameter of about 6 inches would be a good place to start. Maybe I will start out with a 6 inch coil for both the transmitter and the receiver. If that works out, I could then consider reducing the size of one or both of the coils.

And after many experiments we will choose 4 inch coil for the transmitter with 45uH and 6 inch coil the receiver 5 mH.

The Transmitters



The Receiver



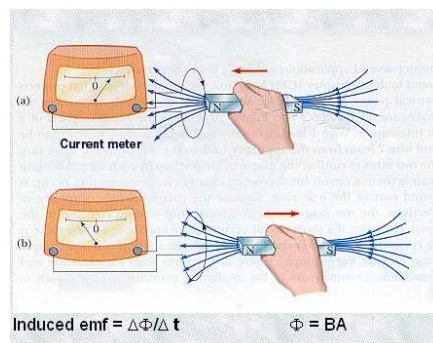
Mutual inductance

*The ring will transmit by Mutual inductance and Inductive coupling

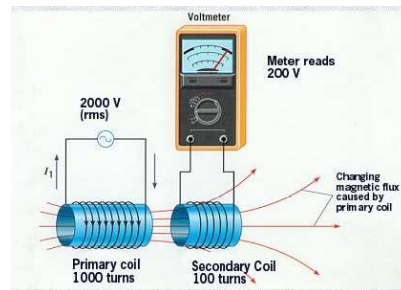
Faraday's law tells us that a changing magnetic flux will induce an emf in a coil. The induced emf for a coil with N loops is:

$$\text{Induced emf} = N \Delta \Phi / \Delta t$$

- Picture two coils next to each other, end to end. If the first coil has a current going through it, a magnetic field will be produced, and a magnetic flux will pass through the second coil. Changing the current in the first coil changes the flux through the second, inducing an emf in the second coil. This is known as **mutual inductance**, inducing an emf in one coil by changing the current through another

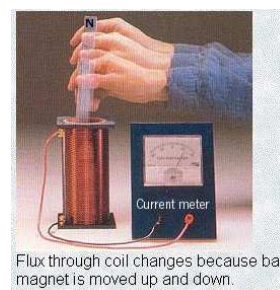
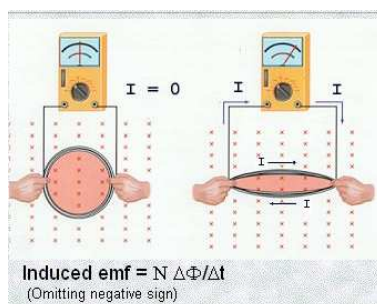


*The induced emf is proportional to the change in flux, which is proportional to the change in current in the first coil. The induced emf can thus be written as:



*The constant M is the mutual inductance, which depends on various factors, including the area and number of turns in coil 2, the distance between the two coils, the relative orientation of the two coils, the number of turns / unit length in the first coil (because that's what the magnetic field produced by the first coil depends on),

and whether the two coils have cores made from ferromagnetic material. In other words, M is rather complicated. What's far more important in the equation above is that the emf induced in the second coil is proportional to the change in current in the first.



Inductive coupling

is the transfer of energy from one circuit (such as a conductive antenna and associated circuitry) to another by means of mutual inductance between the two circuits.

Data Encoding Scheme?

To keep the transmitter power consumption low, I need a data encoding scheme that uses the minimum number of magnetic field pulses. One such method is a pulse position scheme. The method needs only $N + 1$ pulses per data channel.

So to send one in the code I must send two pulses of one (two rings) from the transmitter and to send zero in the code I must send one pulse of one (one ring) from the transmitter for more accuracy and to make sure to recover pulses of one clearly and correctly in the receiver and also pulses of zero.

And I must also include some minimum time (zero level) between the reference pulse and the data pulse so that I must not send two pulses of one continuously to avoid overlapping two pulses of one and to make sure to recover pulses of one clearly in the receiver and also recover pulses of zero; and this is also because that the single drive pulse launched into the LC network will need to be one half the period of 175KHz or 2.86μs and this is the time to produce a pulse of one (ring) and this time is too small so I will send a zero pulse between two pulses of one's to make sure that I can recover them correctly.

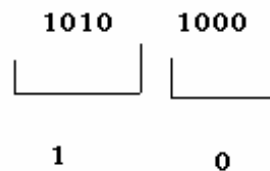
*And I will send the pulses with frequency (20 Hz) to produce a ring with (175 KHz)..

***We select the code of this car is (10) :**
But the the basic code that will send by the microprocessor is :

1010 1000

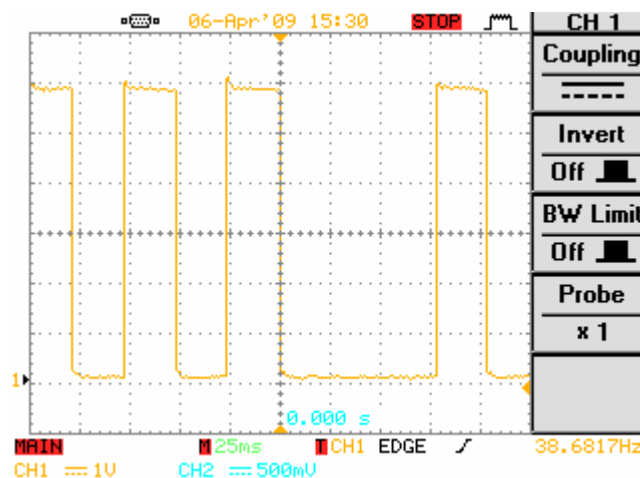
So that:

1010 ----- for 1



1000 ----- for 0

***and my code after the amplifier is shown below:**



**and the programming code of the Arduino that producing our code is:*



```
int ledPin = 12;                // LED connected to digital pin 12

void setup()                    // run once, when the sketch starts
{
  pinMode(ledPin, OUTPUT);      // sets the digital pin as output
}

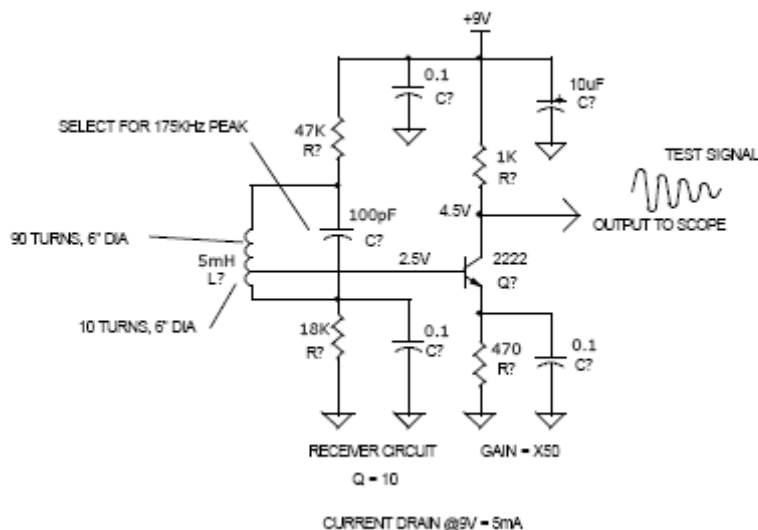
void loop()                     // run over and over again
{
  digitalWrite(ledPin, HIGH);   // sets the LED on
  delay(25);                     // waits for a second
  digitalWrite(ledPin, LOW);    // sets the LED off
  delay(25);                     // waits for a second
  digitalWrite(ledPin, HIGH);   // sets the LED on
  delay(25);
  digitalWrite(ledPin, LOW);    // sets the LED off
  delay(25);
  digitalWrite(ledPin, HIGH);   // sets the LED on
  delay(25);
  digitalWrite(ledPin, LOW);    // sets the LED off
  delay(25);
  digitalWrite(ledPin, LOW);    // sets the LED off
  delay(25);
  digitalWrite(ledPin, LOW);    // sets the LED off
  delay(25);
}
```


5) Ring receiver :

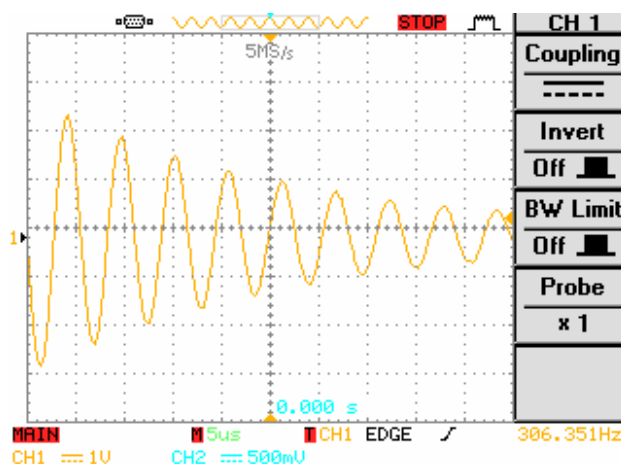
I need to design and build a matching receiver.

For starters, I could use a simple parallel resonant LC network as the receiver circuit. It would be nice if I could get a receive signal in excess of several volts peak to peak at a range of 4 feet. Such a signal would minimize the needed amplifier gain and would allow a receiver circuit to operate in a non-linear mode.

Now , the receiver circuit

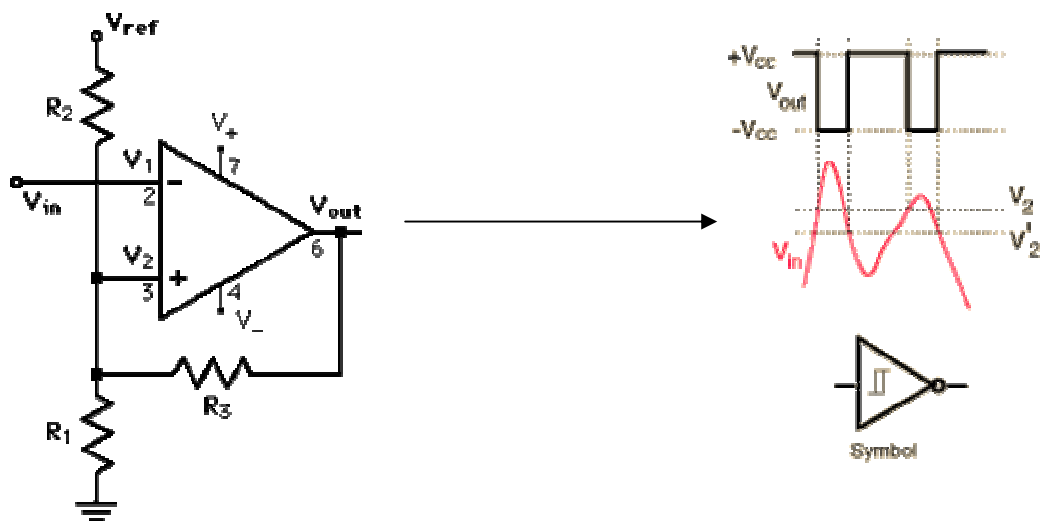


*and the code attach to the receiver will produce a ring for every pulse of one like that:



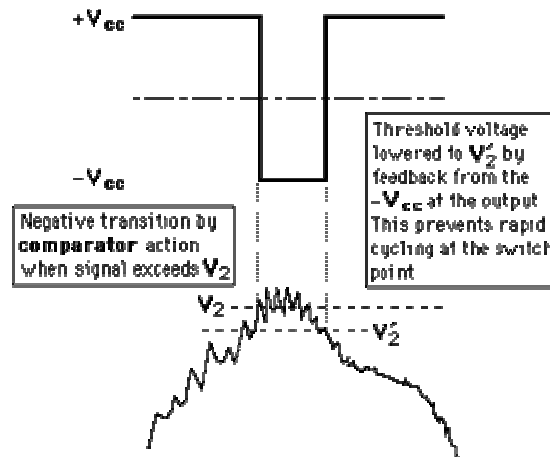
6) Schmitt triggers (pulse generators):

The Schmitt trigger is a comparator application which switches the output negative when the input passes upward through a positive reference voltage. It then uses negative feedback to prevent switching back to the other state until the input passes through a lower threshold voltage, thus stabilizing the switching against rapid triggering by noise as it passes the trigger point.



Schmitt Trigger Action

The Schmitt trigger action uses a comparator to produce stable level-crossing switches in contrast to the action of a straight reference comparison.



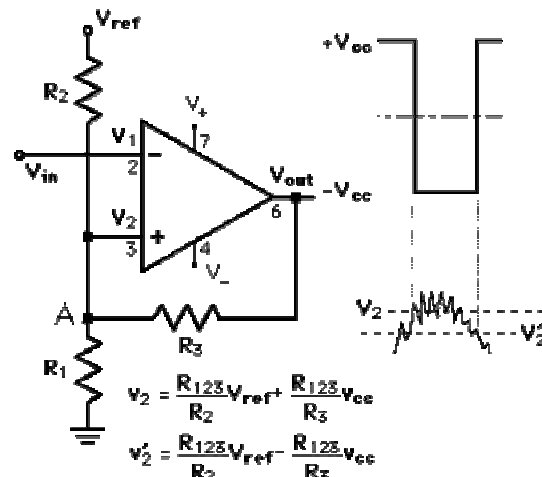
*Schmitt trigger action is a double threshold comparator process. The current equation at A gives:

$$\frac{V_{ref} - V_2}{R_2} + \frac{V_{out} - V_2}{R_3} = \frac{V_2}{R_1}$$

$$V_2 = \frac{R_{123} V_{ref}}{R_2} + \frac{R_{123} V_{out}}{R_3}$$

where $R_{123} = R_1 \parallel R_2 \parallel R_3$

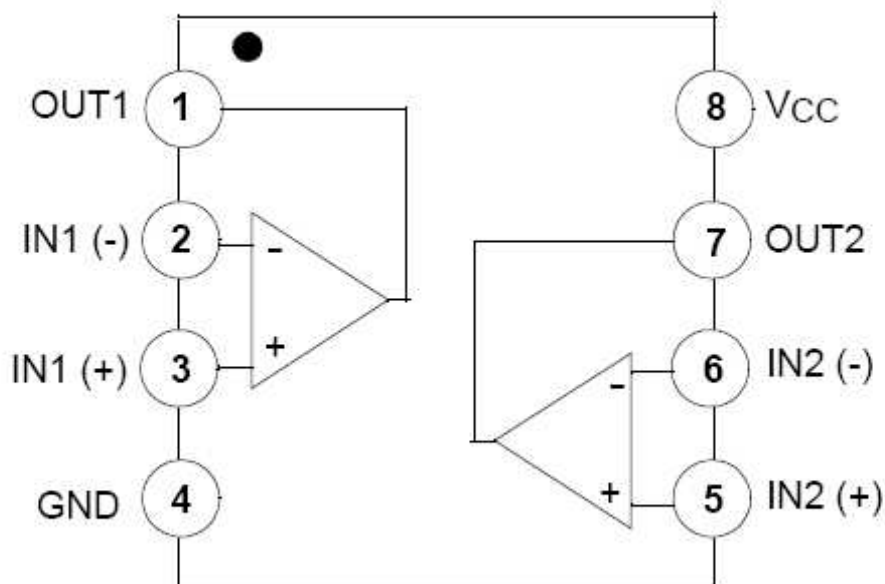
This dependence upon the output voltage gives the dual threshold. The two output states give the thresholds shown at right



Description:

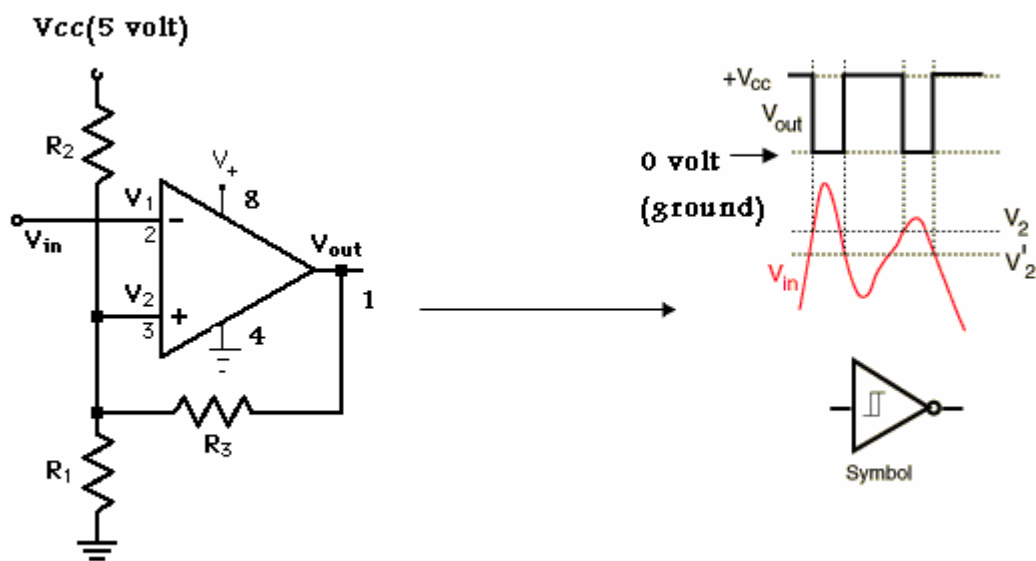
The LM2904, LM358/LM358A, LM258/LM258A consist of two independent, high gain, internally frequency compensated operational amplifiers which were designed specifically to operate from a single power supply over a wide range of voltage. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of the power supply voltage.

Application areas include transducer amplifier, DC gain blocks and all the conventional OP-AMP circuits which now can be easily implemented in single power supply systems

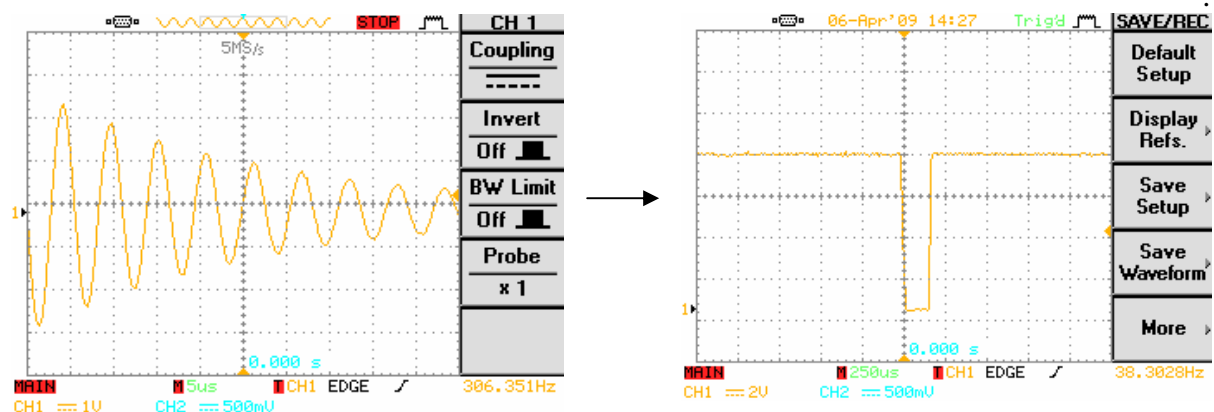


*The result of this stage:

At the end of this stage every ring will produce one negative pulse between (+Vcc and Vcc),but ,we use LM358 as a Schmitt trigger ,and this comparator is work with single power supply (just +Vcc and 0volt),so the output pulse of this Schmitt trigger has an amplitude between (+Vcc and 0 volt) with Vcc=5volts.



*And the result show below:



7) 555 Monostable

Introduction

The 8-pin 555 timer must be one of the most useful ICs ever made and it is used in many projects. With just a few external components it can be used to build many circuits, not all of them involve timing!

A popular version is the NE555 and this is suitable in most cases where a '555 timer' is specified. The 556 is a dual version of the 555 housed in a 14-pin package, the two timers (A and B) share the same power supply pins. The circuit diagrams on this page show a 555, but they could all be adapted to use one half of a 556.

Low power versions of the 555 are made, such as the ICM7555, but these should only be used when specified (to increase battery life) because their maximum output current of about 20mA (with a 9V supply) is too low for many standard 555 circuits. The ICM7555 has the same pin arrangement as a standard 555.

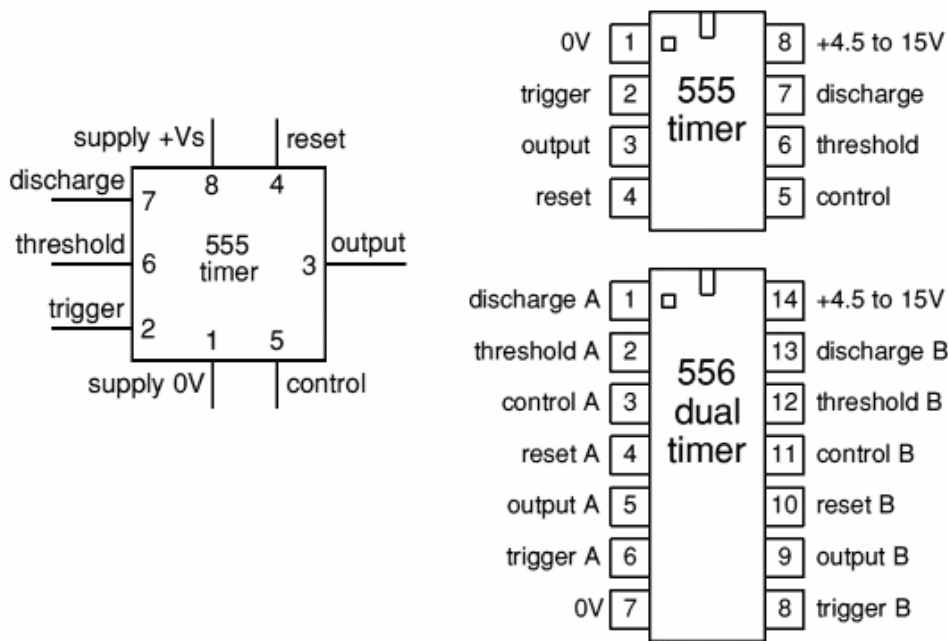
The circuit symbol for a 555 (and 556) is a box with the pins arranged to suit the circuit diagram: for example 555 pin 8 at the top for the +Vs supply, 555 pin 3 output on the right. Usually just the pin numbers are used and they are not labelled with their function.

The 555 and 556 can be used with a supply voltage (Vs) in the range 4.5 to 15V (18V absolute maximum).

Standard 555 and 556 ICs create a significant 'glitch' on the supply when their output changes state. This is rarely a problem in simple circuits with no other ICs, but in more complex circuits a **smoothing capacitor** (eg 100μF) should be connected across the +Vs and 0V supply near the 555 or 556.

The input and output pin functions are described briefly below and there are fuller explanations covering the various circuits:

- Astable - producing a square wave
- Monostable - producing a single pulse when triggered
- Bistable - a simple memory which can be set and reset
- Buffer - an inverting buffer (Schmitt trigger)



Inputs of 555/556

Trigger input: when $< \frac{1}{3} V_s$ ('active low') this makes the output high (+Vs). It monitors the discharging of the timing capacitor in an astable circuit. It has a high input impedance $> 2M\Omega$.

Threshold input: when $> \frac{2}{3} V_s$ ('active high') this makes the output low (0V)*. It monitors the charging of the timing capacitor in astable and monostable circuits. It has a high input impedance $> 10M\Omega$.

* providing the trigger input is $> \frac{1}{3} V_s$, otherwise the trigger input will override the threshold input and hold the output high (+Vs).

Reset input: when less than about 0.7V ('active low') this makes the output low (0V), overriding other inputs. When not required it should be connected to +Vs. It has an input impedance of about $10k\Omega$.

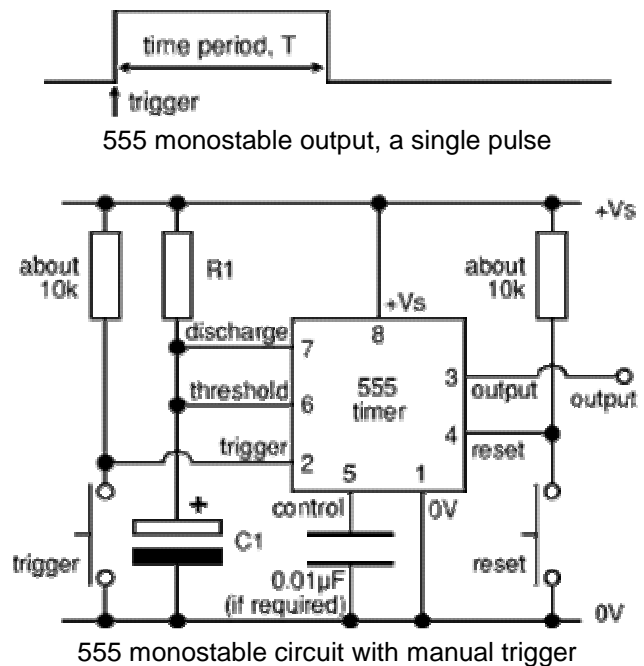
Control input: this can be used to adjust the threshold voltage which is set internally to be $\frac{2}{3} V_s$. Usually this function is not required and the control input is connected to 0V with a $0.01\mu F$ capacitor to eliminate electrical noise. It can be left unconnected if noise is not a problem.

The **discharge pin** is not an input, but it is listed here for convenience. It is connected to 0V when the timer output is low and is used to discharge the timing capacitor in astable and monostable circuits.

555/556 Monostable

A monostable circuit produces a single output pulse when triggered. It is called a monostable because it is stable in just **one** state: 'output low'. The 'output high' state is temporary.

The duration of the pulse is called the **time period** (T) and this is determined by resistor R1 and capacitor C1:



$$\text{time period, } T = 1.1 \times R1 \times C1$$

T = time period in seconds (s)

R1 = resistance in ohms (Ω)

C1 = capacitance in farads (F)

The maximum reliable time period is about 10 minutes.

Why 1.1? The capacitor charges to $\frac{2}{3} = 67\%$ so it is a bit longer than the time constant ($R1 \times C1$) which is the time taken to charge to 63%.

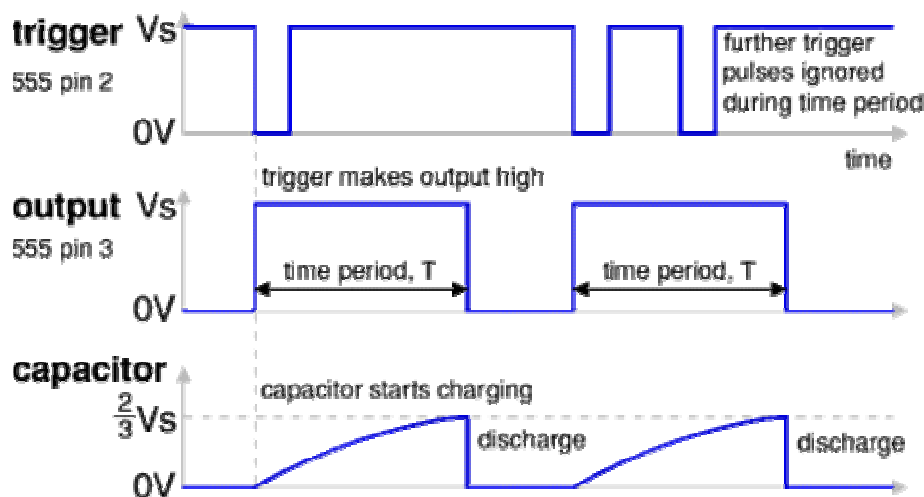
- **Choose C1 first** (there are relatively few values available).
- **Choose R1** to give the time period you need. R1 should be in the range $1k\Omega$ to $1M\Omega$, so use a fixed resistor of at least $1k\Omega$ in series if R1 is variable.
- **Beware** that electrolytic capacitor values are not accurate, errors of at least 20% are common.
- **Beware** that electrolytic capacitors leak charge which substantially increases the time period if you are using a high value resistor - use the formula as only a very rough guide!

Monostable operation

The timing period is triggered (started) when the **trigger** input (555 pin 2) is less than $\frac{1}{3} V_s$, this makes the **output** high ($+V_s$) and the capacitor C1 starts to charge through resistor R1. Once the time period has started further trigger pulses are ignored.

The **threshold** input (555 pin 6) monitors the voltage across C1 and when this reaches $\frac{2}{3} V_s$ the time period is over and the **output** becomes low. At the same time **discharge** (555 pin 7) is connected to 0V, discharging the capacitor ready for the next trigger.

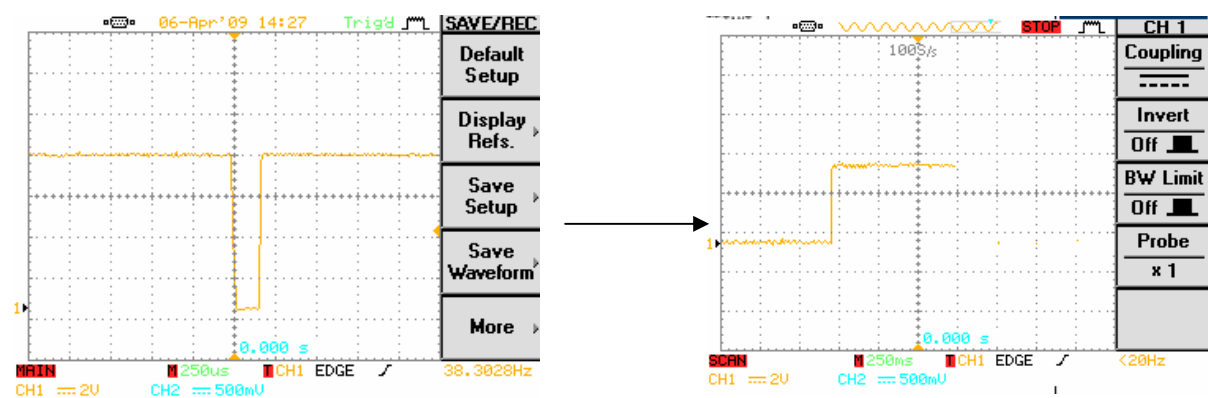
The **reset** input (555 pin 4) overrides all other inputs and the timing may be cancelled at any time by connecting reset to 0V, this instantly makes the output low and discharges the capacitor. If the reset function is not required the reset pin should be connected to $+V_s$.



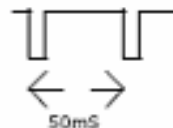
***the result of this stage:**

When the pulse of the Schmitt trigger come to 555 Monostable ,then every pulse will open a time window from the output of 555 Monostable ,and The duration of time of this window called the time period (T) and this is determined by resistor R1 and capacitor C1:

time period, $T = 1.1 \times R1 \times C$



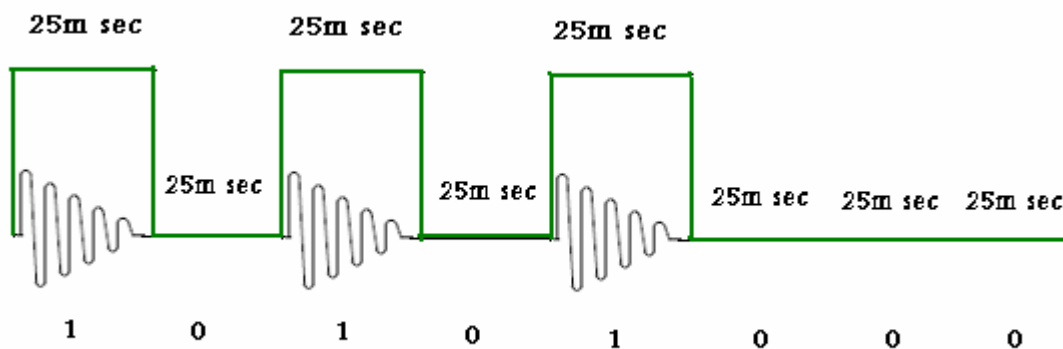
***And we put the time as (25m sec),this is because that one period which produce one ring by the transmitter is take (50m sec).**



***so we will return every ring pulse(one's pulse) in the receiver to pulse of time window with time 25m sec for one's pulse and the rest 25m sec for zero's pulse.**

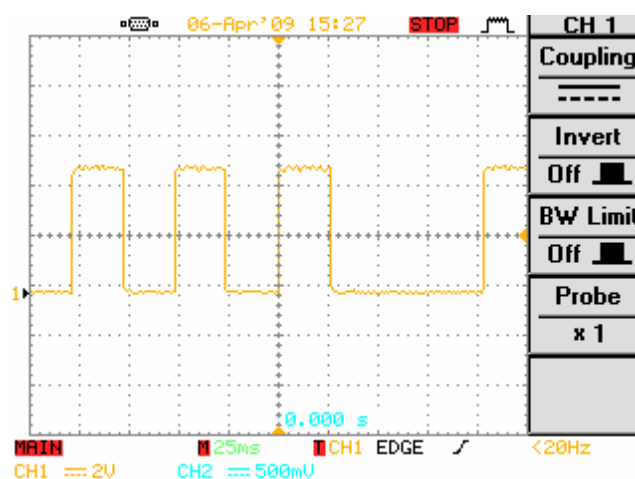
***At the end we will return the code in the receiver to the original code come from the microprocessor in the transmitter with time 25m sec for one's pulse and the rest 25m sec for zero's pulse.**

As show



***And the code which we return is**

1010 1000



8)Microprocessor of Receiver (Arduino) :



***At the end the code come to the microprocessor in the receiver is (10101000) but we must return the original code which transmit by the microprocessor in the transmitter which is (10) and this is the correct code of the car which we must return.**

***so with programming the Arduino we will take the code which come from monostable and write a program to do this steps:**

-when the code come to microprocessor the program will open a time window with time (90m sec) to check every four bits.

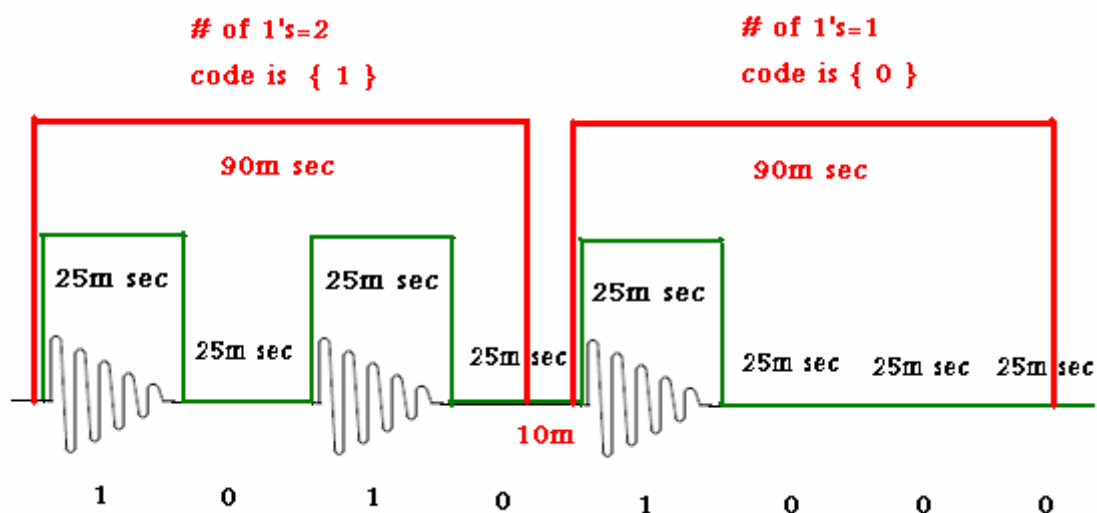
-then the program will count the number of one's pulses in this time(90m sec).

-if number of one's pulses in this time(90m sec) equal two ,then the microprocessor convert this four bits to one pulse of one (5 volts)

- if number of one's pulses in this time(90m sec) equal one ,then the microprocessor convert this four bits to one pulse of zero (0 volts)

-after every open of 90m sec(check four bits) ,the program will make a delay of (10m sec) then start to open other time window(90m sec) {start to check the next four bits}.

***And for our code is shown below:**



***At the end of the receiver the Microprocessor will take the code (10) and then check if this code was one of the existing codes which are allowed to enter the car park, if the car was carrying the code which is allowed to enter then the receiver give an order to open the gate(door) of the car park to the car, and if the code was not part of this codes (codes which are allowed to enter the gate) then the door remains closed and the car don't enter the car park(Garage).**

*The code of the microprocessor in the receiver to check the code:

```

int startbits[] = {
    1, 1, 1};
int comingstartbits[] = {
    0,0,0};
int databits[] = {
    0, 0, 0, 0, 0, 0};
int comingdatabits[] = {
    0,0,0,0,0,0};
int coming = 0;
int x =0;
int check = 0;
void setup(){
    pinMode(13,OUTPUT);
    pinMode(12,INPUT);
    pinMode(10,OUTPUT);
}

void loop(){
    int y=0;
    for(int j=0; j<=2; j++){
        check = digitalRead(12);

        if (check == HIGH)
            y++ ;
        delay(30);
        coming = digitalRead(12);
        if (check == HIGH){
            y++ ;
        }
        delay(20);
        check= digitalRead(12);

        if (check == HIGH){
            y++ ;
        }
        delay(30);
        check = digitalRead(12);

        if (check == HIGH){
            y++ ;
        }
        delay(10);
        delay(10);
        if (y==2){
            comingstartbits[j] = 1;
        }
        if (y==1){
            comingstartbits[j] =0;
        }
    }

    if (comingstartbits = startbits){
        x=0;
        for(int i=0; i<=4 ; i++){
            check = digitalRead(12);

            if (check == HIGH)
                x++ ;
            delay(30);
            check = digitalRead(12);

            if (check == HIGH){
                x++ ;
            }
            delay(20);
            check = digitalRead(12);

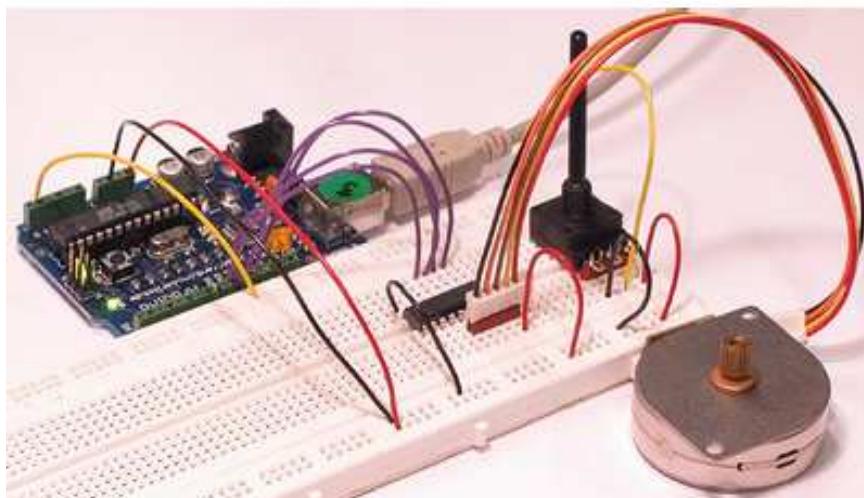
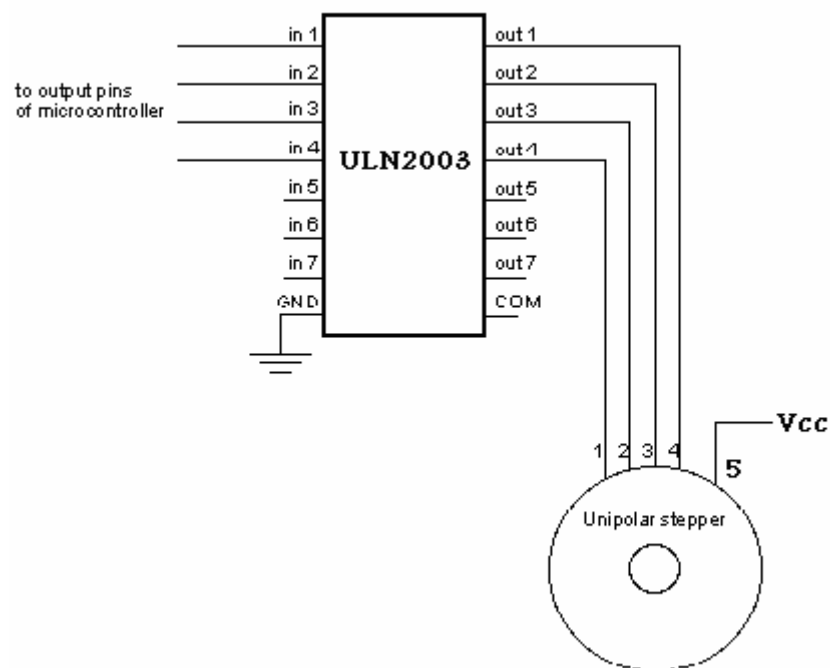
            if (check == HIGH){
                x++ ;
            }
            delay(10);
            delay(10);

            if (x==2){
                comingdatabits[i] == 1;
            }
            else if (x==1){
                comingdatabits[i] == 0 ;
            }
        }
        for(int k=0 ; k<=4; k++){
            digitalWrite(10,comingdatabits[k]);
            delay(2000);
        }
        if (comingdatabits = databits){
            digitalWrite(13,comingdatabits[0]);
            delay(5000);
        }
    }
}

```

9)Stepper Motor (to open the door of the car park) :

If the code of the car is correct and the receiver check if this code was one of the existing codes which are allowed to enter the car park , then the microprocessor give an order to the stepper motor that control with the door of the car perk, then the stepper motor will rotate and open the door with angle (90) the make a delay for example (5 sec) ,then the stepper will rotate to close the door with angle (-90).



*The code of the stepper to control the door:

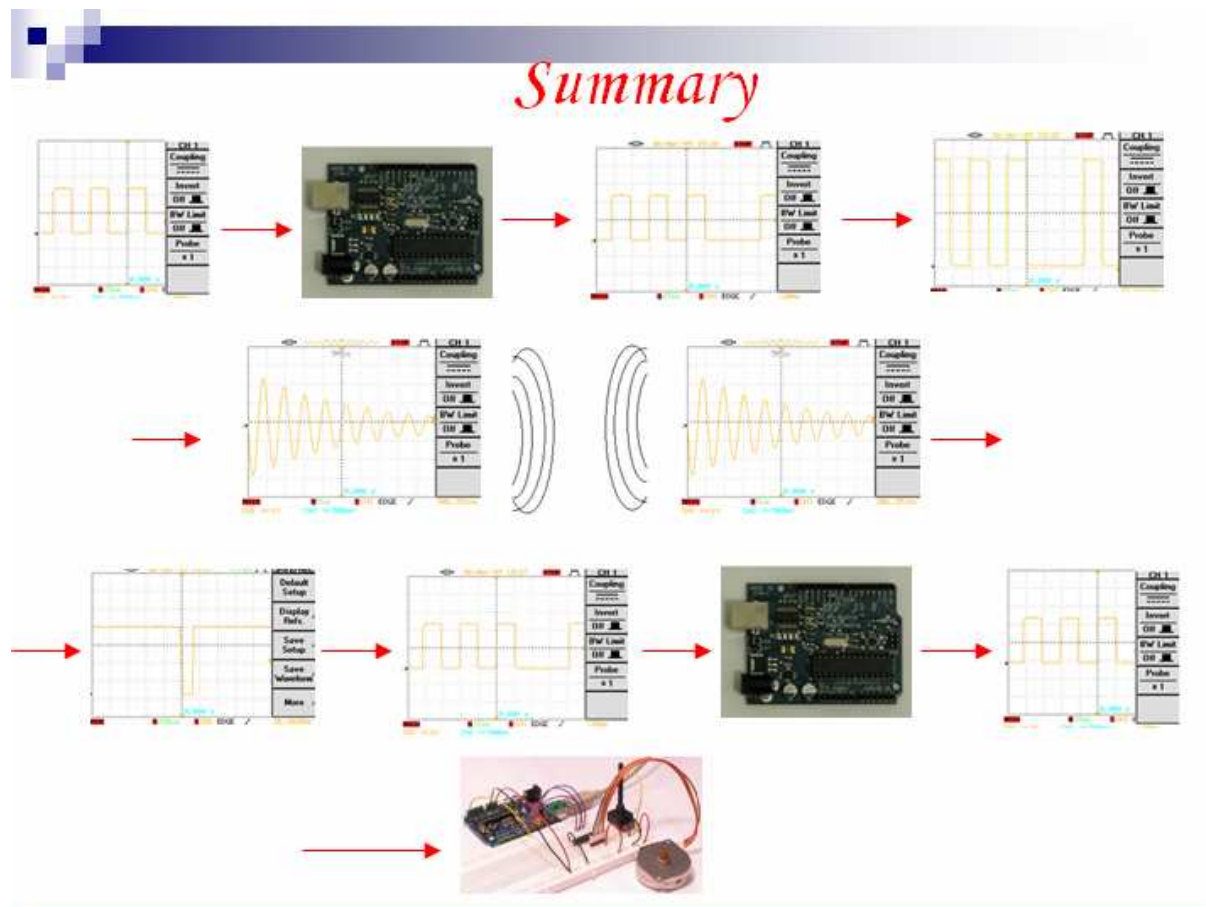
```
int motorPin1 = 8;
int motorPin2 = 9;
int motorPin3 = 10;
int motorPin4 = 11;
int delayTime = 80;

void setup() {
    pinMode(motorPin1, OUTPUT);
    pinMode(motorPin2, OUTPUT);
    pinMode(motorPin3, OUTPUT);
    pinMode(motorPin4, OUTPUT);

    digitalWrite(motorPin1, HIGH);
    digitalWrite(motorPin2, LOW);
    digitalWrite(motorPin3, LOW);
    digitalWrite(motorPin4, LOW);
    delay(delayTime);
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, HIGH);
    digitalWrite(motorPin3, LOW);
    digitalWrite(motorPin4, LOW);
    delay(delayTime);
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, LOW);
    digitalWrite(motorPin3, HIGH);
    digitalWrite(motorPin4, LOW);
    delay(delayTime);
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, LOW);
    digitalWrite(motorPin3, LOW);
    digitalWrite(motorPin4, HIGH);
    delay(delayTime);
    digitalWrite(motorPin1, HIGH);
    digitalWrite(motorPin2, LOW);
    digitalWrite(motorPin3, LOW);
    digitalWrite(motorPin4, LOW);
    delay(delayTime);
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, HIGH);
    digitalWrite(motorPin3, LOW);
    digitalWrite(motorPin4, LOW);
    delay(delayTime);
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, LOW);
    digitalWrite(motorPin3, HIGH);
    digitalWrite(motorPin4, LOW);
    delay(delayTime);
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, LOW);
    digitalWrite(motorPin3, LOW);
    digitalWrite(motorPin4, HIGH);
    delay(delayTime);
}

void loop() {
```

**Summary in pictures :*



The End