Skype-controlled robot using smartphone and DTMF tones

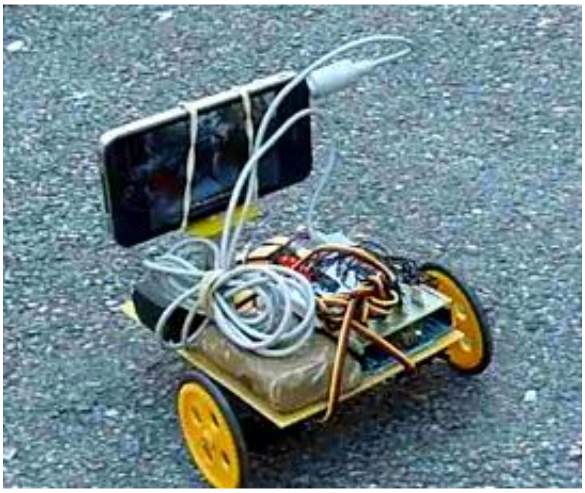
http://www.instructables.com/id/Skype-controlled-robot-using-smartphone-and-DTMF-t/

A new trend in robotics is smartphones that have a lot of sensors, camera, GPS. One of problems, however, is how to interface phone with custom electronics. It can be done over Bluetooth, but audio interfacing using dual tone modulation frequency DTMF is simpler. That is enough to control a simple robot described here.

It is a simple moving platform containing mobile phone that is remote-controlled over Internet via Skype using DTMF tones. Wireless connection is done by WLAN that is free of charge or mobile phone network that can be done over large distance.

Here is a video where Skype video call is started from a PC to iPhone. A dial pad is opened in Skype to generate DTMF tones. Video at

http://www.youtube.com/watch?v=2-mUVKr5XLI



Dual frequency DTMF method is quite reliable and there exists a specialized decoder chip. DTMF chip is read out by Arduino board that controls servos.

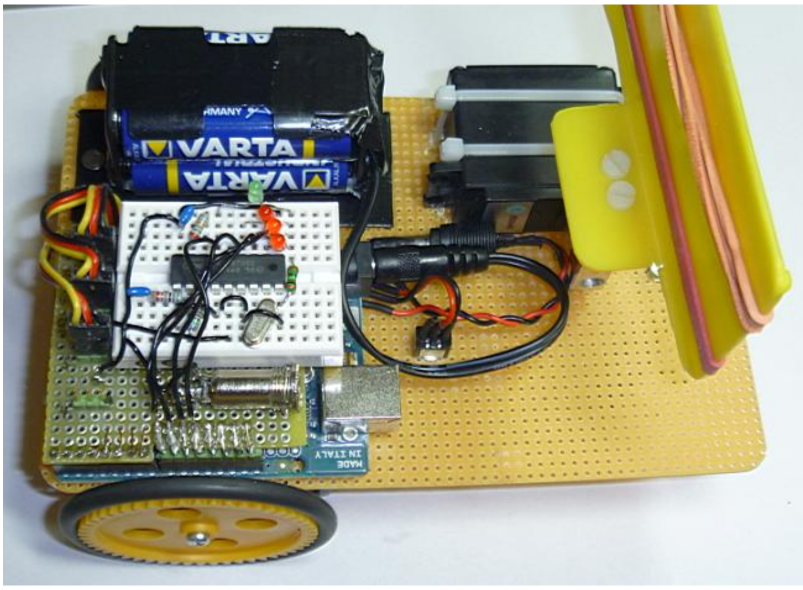
A very similar robot controlled over mobile phone, but with motors instead for servos, is very well described in Instructable:

http://www.instructables.com/id/Cellphone-Operated-Robot-1/step3/SCEMATIC-OF-CELLPHONE-OPERATED-LANDROVER/

Here are some robotics sites using smartphones:
http://www.cellbots.com/
http://code.google.com/p/cellbots/
https://github.com/mleone/broadcast
http://www.razorconcepts.net/dtmf.html

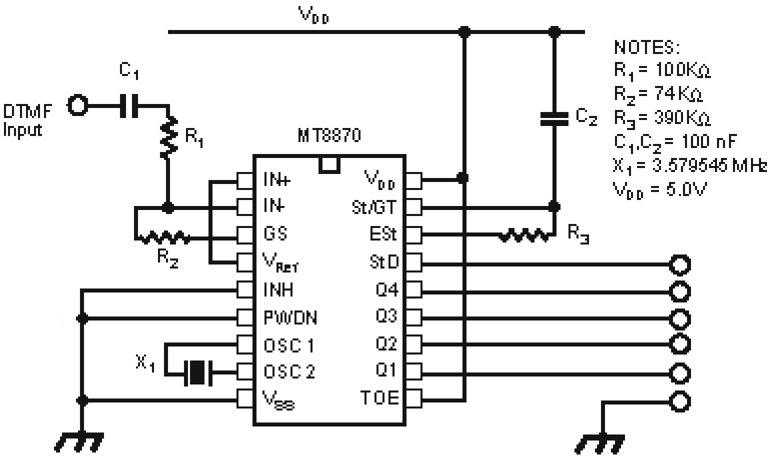
Construction

Mechanical construction is nothing particular and can be found by searching google for simple robots:
http://www.instructables.com/id/How-to-Make-an-Arduino-Controlled-Servo-Robot-SER/
Servos are like used in radio-controlled models but are modified for continuous rotation:
http://www.instructables.com/id/Modify-A-Servo-For-Continuous-Rotation/



I will concentrate on electronics here.

When a button is pressed in Skype, a dual-tone audio signal is generated that is sent by a wireless connection to a phone that plays it. DTMF decoder chip MT8870 input is connected to headphone output of iPhone, or some other smart(phone). In principle a microphone could be used to skip the audio cable.



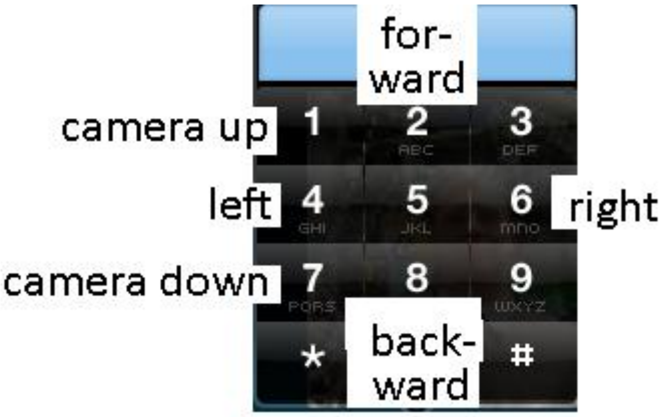
Each number is coded by two frequencies. DTMF chip decodes frequencies and sets lines Q1-Q4 according to the table in manufacturer datasheet. Decoding is easy as the code is actually the binary representation of the number. LEDs can be connected to the DTMF chip pins for visual checking.

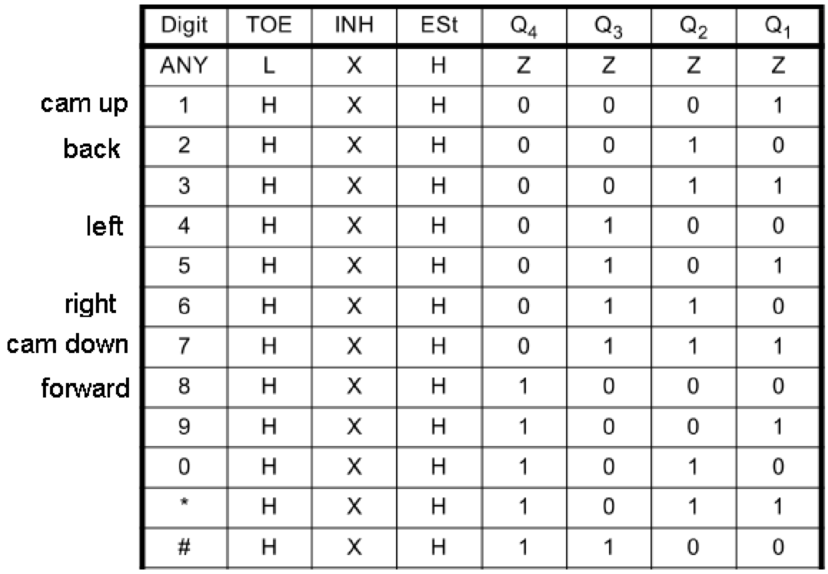
StD goes High when valid digit is detected. It is used for checking that a command has been sent and is connected to Arduino pin 8.

Lines Q1, Q2, Q3 are connected directly to Arduino digital input pins 9, 10, 11. This is Arduino port B that can be read out by a single command “PINB”, see programm code in appendix. Arduino boards have very good documentation from developer and in forums. The pin numbers of Arduino are not the pin numbers of Atmega chip. Arduino programming software usees Arduino pin numbers and not the chip pin numbers.

Arduino board reads a byte form Port B input lines and if a correct combination exists, then executes a subroutine for servo move. For Arduino exists a ready-made servo control library. Wheel servos are modified for continuous rotation. Arduino code also allows to steer robot over a COM-port interface by sending numbers from PC keyboard.

Below is a Skype dial pad and buttons that are used to steer the robot.
Dialpad is under /call/ show/dialpad. On smaartphones exist DTMF tone apps.
.

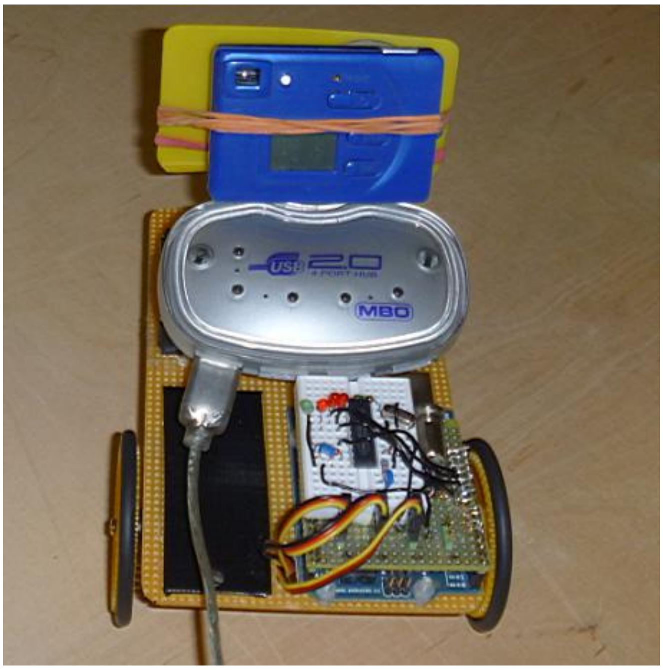




	Digit	TOE	INH	ESt	Q_4	Q_3	Q_2	Q_1
	ANY	L	X	H	Z	Z	Z	Z
cam up	1	H	X	H	0	0	0	1
back	2	H	X	H	0	0	1	0
	3	H	X	H	0	0	1	1
left	4	H	X	H	0	1	0	0
	5	H	X	H	0	1	0	1
right	6	H	X	H	0	1	1	0
cam down	7	H	X	H	0	1	1	1
forward	8	H	X	H	1	0	0	0
	9	H	X	H	1	0	0	1
	0	H	X	H	1	0	1	0
	*	H	X	H	1	0	1	1
	#	H	X	H	1	1	0	0

Power

Batteries attached with help of magnet. Be carefull with charging LiPo. If voltage increases above 4.2 V/ cell then cell might ignite. Use only special chargers like for radio controlled models. I have 2 series LiPo pack. For charging I connect cells in parallel and charge by putting inside mobile phone a dummy with contacts.



PC webcam drive around

For autonomous operation battery is the biggest problem. For around a clock operation connect with USB cable to PC. Use webcam and USB hub. Now you can skype to it any time.

Remarks after testing

The Skype robot project was quite fun and it would be nice to have it online all the time. For that one would need to solve power supply problem. May be a solar panel, and put it in low consumption sleep mode when nobody controls it.

Android smartphones at the moment of writing did not have video in Skype. Need visual contact or an app Droidcam to see video in WLAN.

For a Skype call from an iPhones no dial tone pad was available at the moment of writing.

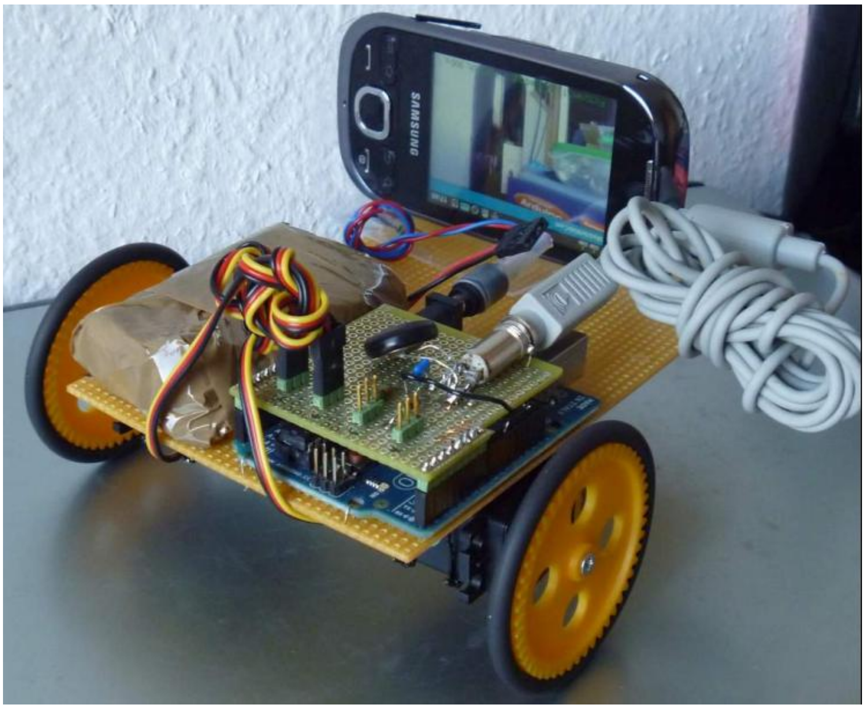
DTMF commands are ca 200 ms long, so do not expect to be able to control the robot rapidly.

If wished one could make all commands endless in time and introduce a stop button as nr 5.

```
//  ROBOT controlled via INTERNET   Skype > DTMF decoder > Arduino > servos
#include <Servo.h>
 Servo myservol, myservor, myservoc;
 int c;  //command
 int pos = 117; // position of camera tilt servo

void setup()
{
  DDRB = B00000000;  // sets Arduino pins 8-13 as inputs
  myservol.attach(7);  myservor.attach(6); myservoc.attach(5);
  myservol.write(90);  myservor.write(90); myservoc.write(pos); //stop motion and put camera looking
horizontally

  Serial.begin(9600);
  while(Serial.available())  Serial.read();

  turnright(); delay(200); turnleft(); delay(200); stopp(); // make some movements on power up
  Serial.println ("commands via serail 4-left, 6-right, 2-forw, 8-back, 1-cam up, 7-cam down. Or DTMF
tones via audio");
}
void forward()  { myservol.write(10);  myservor.write(170); delay(1000); }
void backward() { myservol.write(170); myservor.write(10);  delay(1000); }
void turnright(){ myservol.write(10);  myservor.write(10);  delay(100);  }
void turnleft() { myservol.write(170); myservor.write(170); delay(100);  }
void camup()    { pos=pos+3; Serial.println(pos,DEC); if (pos>180){pos=180;} myservoc.write(pos);
delay(300);}
void camdown()  { pos=pos-3; Serial.println(pos,DEC); if (pos< 90){pos= 90;} myservoc.write(pos);
delay(300);}
void stopp()    { myservol.write(90);  myservor.write(90); }

void loop()
{
  if (Serial.available())
  {  c = Serial.read();  Serial.println(c);
     if (c==50) { forward();  }
     if (c==56) { backward(); }
     if (c==52) { turnleft(); }
     if (c==54) { turnright();}
     if (c==55) { camup(); }
     if (c==49) { camdown();}
     stopp();
  }
  c=PINB;
  if(c==8) {backward();}
  if(c==10) {forward();}
  if(c==12) {turnleft();}
  if(c==14){turnright();}
  if(c==15) {camup();}
  if(c==9){camdown();}
  stopp();
}
```

Single-tone control
Unsuccessful project.

Robot control using single tone was tried prior to DTMF, however, it was not very reliable and software frequency measurement took a lot of time. Some notations in the program code still contain traces of this initial attempt.



First version was to use single frequency tones 700, 800, 900, 1000, 1100, 1200 Hz.
The tones were recorded by *Cooledit* as wave files 1 s long.
Skype can transmit tones well in this range. Arduino had a program that counts frequency.

http://tushev.org/articles/electronics/43-measuring-frequency-with-arduino

```
long getFrequency(int pin) {
#define SAMPLES 4096
long freq = 0;
for(unsigned int j=0; j<SAMPLES; j++) freq+= 500000/pulseIn(pin, HIGH,
250000);
return freq / SAMPLES;
```

A webpage was made with buttons playing tones. Audio input and output on PC were linked with cable.

```
<html> <body> <span id=dummyspan></span>
<form>
<input type="button" value="Forward" onClick="DHTMLSound('700.wav')">
<input type="button" value="Backward" onClick="DHTMLSound('800.wav')">
<input type="button" value="Left" onClick="DHTMLSound('900.wav')">
<input type="button" value="Right" onClick="DHTMLSound('1000.wav')">
</form>
<script>
function DHTMLSound(surl) {
document.getElementById("dummyspan").innerHTML=
"<embed src='"+surl+"' hidden=true autostart=true loop=false>";
}
</script> </body> </html>
```