

ARCH 879/ECE 893 - Architectural Robotics

mKare: The Mobile Care Unit

By Adam James + K V Krishna Teja

Video: <http://www.youtube.com/watch?v=RqjJb1NtRV4>



Abstract:- The mKare is an interactive mobile unit which is built to aid the physically challenged people in their daily lives. mKare can be controlled by a Wii remote to move in all directions. The sides have arms that rise up to provide an extension whenever necessary, thereby giving more workspace and at the same time keeping the overall size of the table compact. mKare is equipped with a smart lighting system which turns on when there is insufficient ambient light. Overall the mKare is a user-friendly mobile assistant which suits the needs and requirements of the physically challenged people.

Scenario:- Nick Papageorgerio recently suffered from a stroke, and from this, lost his ability to move his left side of his body. After Nick recovered from the initial stage, the doctors dismissed him from the hospital, prescribed him medicine, as well as a new device called the mKare to help him at home. When Nick arrived home and his immediate family left him, he soon realized how important having an assistant would be, and realized why the doctor

prescribed the mKare unit along with medicine. In the morning, Nick turned on his favorite television show and then proceeded to get his daily coffee. He soon realized that controlling the wheelchair and holding a cup of hot coffee with one hand would be impossible. Nick grabbed his control for his mKare and quickly controlled it to where he was located. He placed his coffee in the built in cup holder, and went back to the living room to watch television, and controlled the mKare to follow him. After his television show was over, Nick decided to write a letter to his grandkids, however, his new wheelchair did not allow him to use the furniture currently in his house. He quickly remembered his mKare unit and aligned his wheelchair sensors to the new unit, and a table end emerged from the personal care unit and Nick was able to slide under the table and finish writing his letter.

After a long day, Nick decided it was time for bed and headed for bed. When he got to bed, he grabbed his favorite book and began reading, but his arm was getting tired from holding the book. Nick grabbed his remote and again controlled mKare to his bedside, where he activated a sensor to open the side table so he could have a reading surface for bed. Once the arm was, mKare detected low light in the room, and automatically turned on additional lights inside the room.

After Nick finished reading and went to bed, he woke up in the middle of the night to loud thunder and flashes of lightning. The next loud boom turned off the power in the house, and to Nick's delight, the mKare unit automatically detected the dark light condition, and turned its own lights on to help Nick see.

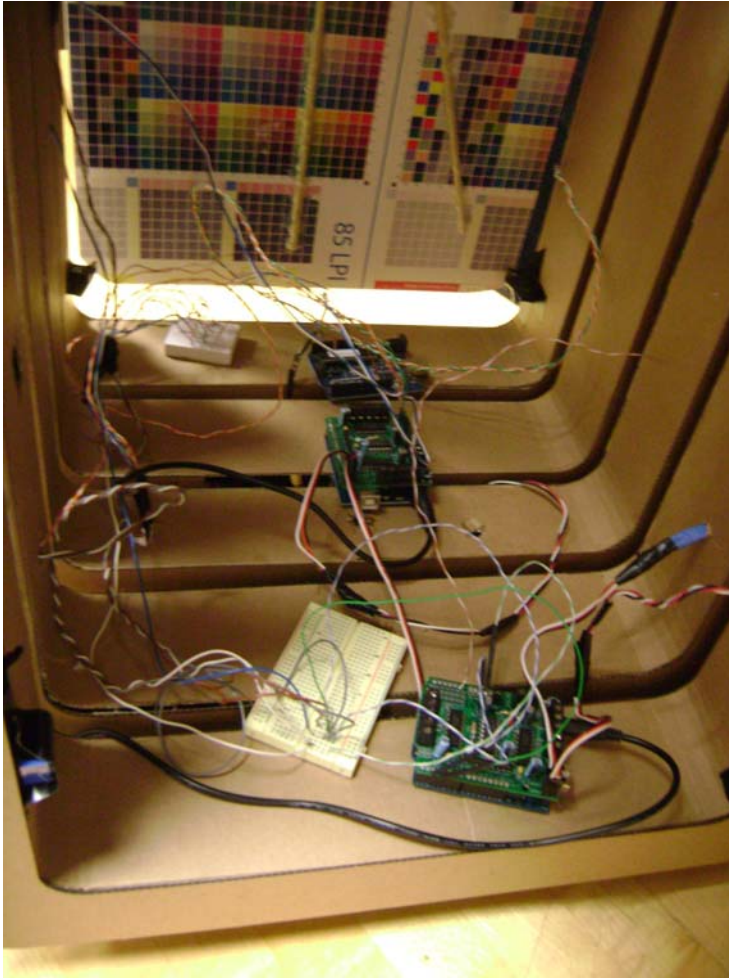
Scenario



mKare is standing by ready to be used.



mKare 's arms have been activated by the user. Both sides are capable of being used if needed.



The inside of mKare.

Hardware requirements

4 RC servomotors

Wii Remote

IR Led's (1)

IR Receivers (2)

PhotoResistor (1)

Resistors (1Kohm-3)

LED's (6)

3 Arduino Boards

Omni directional Wheels (2)

Laser Cut Cardboard

Lego parts.

Code for Wii Control

```
#include <Wire.h>

#include "nunchuck_funcs1.h"

int loop_cnt=0;

byte accx,accy,zbut,cbut;

int servoPin1 = 9;

int servoPin2 = 10;

void setup()

{

  Serial.begin(19200);

  nunchuck_setpowerpins();

  nunchuck_init(); // send the initialization handshake

  Serial.print("WiiChuckDemo ready\n");

  pinMode(servoPin1,OUTPUT);

  pinMode(servoPin2,OUTPUT);

}

void loop()

{

if( loop_cnt > 100 ) { // every 100 msecs get new data

  loop_cnt = 0;

  nunchuck_get_data();
```

```
accx = nunchuck_accelx(); // ranges from approx 70 – 182 in X direction

accy = nunchuck_accely(); // ranges from approx 65 – 173 in y direction

zbut = nunchuck_zbutton();

cbut = nunchuck_cbutton();

Serial.print("accx: "); Serial.print((byte)accx,DEC);

Serial.print("\taccy: "); Serial.print((byte)accy,DEC);

if(accy>=180){

    int temp;

for (temp = 0; temp <=20; temp++)

{

digitalWrite(servoPin1,HIGH);

delayMicroseconds(1400);

digitalWrite(servoPin1,LOW);

delay(20);

}

}

if(accy<=100){

    int temp;

for (temp = 0; temp <=20; temp++)

{

digitalWrite(servoPin1,HIGH);

delayMicroseconds(1600);

digitalWrite(servoPin1,LOW);

delay(20);

}

}

if(accx>=180){
```

```
    int temp;
    for (temp = 0; temp <=20; temp++)
    {
        digitalWrite(servoPin2,HIGH);
        delayMicroseconds(1400);
        digitalWrite(servoPin2,LOW);
        delay(20);
    }
}

    if(accx<=80){
        int temp;
        for (temp = 0; temp <=20; temp++)
        {
            digitalWrite(servoPin2,HIGH);
            delayMicroseconds(1600);
            digitalWrite(servoPin2,LOW);
            delay(20);
        }
    }

    loop_cnt++;
    delay(1);
}
```

Code for Side Arms of the table

```
#include <ServoTimer1.h>

ServoTimer1 servo1;

ServoTimer1 servo2; // variable to store the servo position

int Action1 = 2;

int A1 = 0;

int Action2 = 4;

int A2 = 0;

int servopin1=9;

int servopin2=10;

void setup()

{

  servo1.attach(servopin1) ; // attaches the servo on pin 9 to the servo object

  servo2.attach(servopin2) ; // attaches the servo on pin 10 to the servo object

  pinMode(Action1,INPUT);

  pinMode(Action2,INPUT);

}

void loop()

{

  A1 = digitalRead(Action1);

  servo1.write(170);

  if (A1==LOW) {

  int pos=0;

  for(pos = 0;pos<=90;pos++)

  {

    // in steps of 1 degree
```



```
servo1.write(pos);    // tell servo to go to position in variable 'pos'  
delay(100);          // waits 200ms for the servo to reach the position  
}  
for(pos = 90; pos<=170; pos++)  
{  
servo1.write(pos);  
delay(100);  
}  
}
```

```
A2 = digitalRead(Action2);
```

```
servo2.write(0);
```

```
if (A2==LOW) {  
int pos = 0;  
for(pos = 0;pos<=90;pos++)  
{  
// in steps of 1 degree  
servo2.write(pos);    // tell servo to go to position in variable 'pos'  
delay(200);          // waits 200 ms for the servo to reach the position  
}  
for(pos = 90; pos>=1; pos--)  
{  
servo2.write(pos);  
delay(200);  
}  
}
```

```
}
```

Code for the Lights

```
int photoPin = 2; // select the input pin for the photoresistor
```

```
int ledPin1 = 11; // select the pin for the LED
```

```
int ledPin2 = 12;
```

```
int ledPin3 = 13;
```

```
int val = 0; // variable to store the value coming from the sensor
```

```
void setup() {
```

```
    pinMode(ledPin1, OUTPUT);
```

```
    pinMode(ledPin2, OUTPUT);
```

```
    pinMode(ledPin3, OUTPUT); // declare the ledPin as an OUTPUT
```

```
    Serial.begin (9600); // tells the Arduino to get ready to send data back to the computer
```

```
}
```

```
void loop() {
```

```
    val = analogRead(photoPin); // read the value from the sensor
```

```
    Serial.print ("Value = "); // report the text "Value = "
```

```
    Serial.println (val);
```

```
if(val<=200){
```

```
    digitalWrite(ledPin1, HIGH); // turn the ledPin on
```

```
    digitalWrite(ledPin2, HIGH);
```

```
    digitalWrite(ledPin3, HIGH);
```

```
} // stop the program for some time
```

```
else
```

```
{
```

```
    digitalWrite(ledPin1, LOW); // turn the ledPin
```

```
digitalWrite(ledPin2, LOW);  
  
digitalWrite(ledPin3, LOW);  
  
//delay(val);          // stop the program for some time  
  
}  
  
}
```

Problems Encountered

- 1) Balancing the weight of the table on only two wheels was a challenge as it compromised the smooth movements of the table. This problem can however be very easily solved if we used four wheels controlled by 4 RC servos.
- 2) Power supplied to the Arduino was not sufficient for the RC servos to lift the side arms to their fullest extent