**Data Transfer Using Android Bluetooth using Arduino Uno Development Board**

GROUP MEMBERS:

12BCE1049 PRATIK KAPASI

12BCE1061 ADITYA G KARNAM

12BCE1132 S.ANUPRITHA

**REPORT**

**SOFTWARES USED:**

1) S2 Terminal for Bluetooth

This application is terminal software for communication with the Bluetooth device by using SPP(RFCOMM).

2)Arduino ERW 1.0.5

This is used for compiling ,updating code.This also enables serial commucation between the devices connected.

**INBUILT FUNCTIONS USED:**

1)Serial.read()-Reads the incoming serial data.

2)Serial.available()-Get the number of bytes (characters) available for reading from the serial port. This is data that's already arrived and stored in the serial receive buffer (which holds 64 bytes).

3)Serial.begin()-Sets the data rate in bits per second (baud) for serial data transmission.(For our project we have set the baud rate as 115200).

4)Serial.print()-Prints data to the serial port as human-readable ASCII text. Numbers are printed using an ASCII character for each digit. Floats are similarly printed as ASCII digits, defaulting to two decimal places. Bytes are sent as a single character. Characters and strings are sent as is. An optional second parameter specifies the base (format) to use; permitted values are BIN ,OCT, DEC, HEX.

5)Serial.println()-Prints data to the serial port as human-readable ASCII text followed by a carriage return character (ASCII 13, or '\r') and a newline character (ASCII 10, or '\n').

**PROGRAMS:**

**FIRST CODE:**

int ledPin = 13; // use the built in LED on pin 13 of the Un

int flag = 1; // make sure that you return the state only once

char d = 0;

void setup()

{

// sets the pins as outputs:

pinMode(ledPin, OUTPUT);

digitalWrite(ledPin, LOW);

Serial.begin(115200);

}

void loop()

{

if(Serial.available()){

if(flag)

{

Serial.println("");

Serial.println("Data transfer has initiated.");

flag=0;

}

d = Serial.read();

Serial.println("");

Serial.print("Received: ");

Serial.println(d);

}

}

**Description** : In this program data transfer operation has been implemented.The data that is given as input by the user is displayed at the S2 terminal as well as the serial communication interface. The data is displayed one character at a time even if string is given as an input .

**Screenshots:**



**2)SECOND CODE:**

int ledPin = 13; // use the built in LED on pin 13 of the Un

int flag = 1; // make sure that you return the state only once

char d = 0, i=0;

char BUFF[100];

void setup()

{

// sets the pins as outputs:

pinMode(ledPin, OUTPUT);

digitalWrite(ledPin, LOW);

Serial.begin(115200);

}

void loop()

{

if(Serial.available()){

if(flag)

{

Serial.println("");

Serial.println("Data transfer has initiated...");

flag=0;

}

d = Serial.read();

if(d=='$')

{

memset(BUFF, 0, 100);

i=0;

}

else

{

BUFF[i] = d;

i++;

if(i==100)

i=0;

Serial.println("");

Serial.print("Text: ");

Serial.println(BUFF);

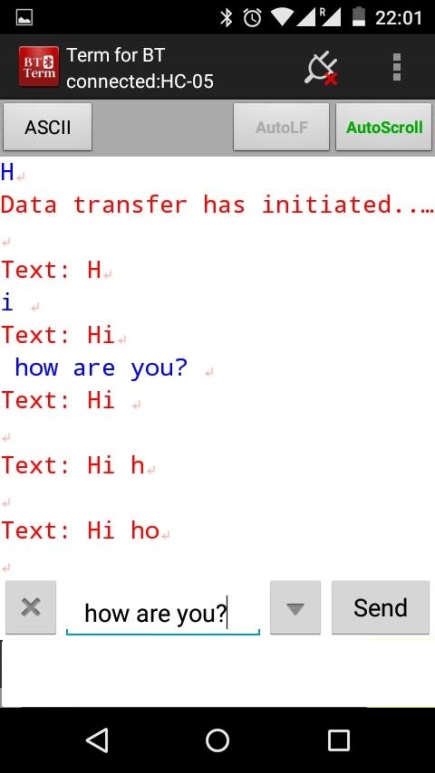
}

}

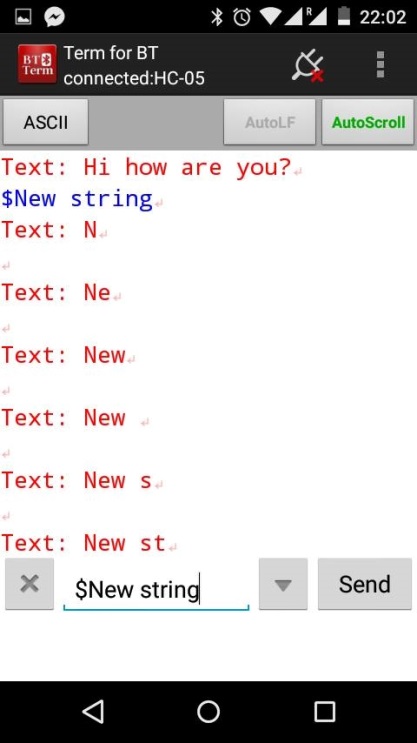
}

**Description** :This code is an extension to the previous code. In this the single characters that are read at a time are stored in a buffer. Here the status of the buffer is being displayed as the output.Also to enter a new string prepend it with a $ symbol.This flushes out the data already stored in the buffer and displays the buffer status of the current string.

**Screenshots:**









**THIRD CODE:**

int ledPin = 13; // use the built in LED on pin 13 of the Uno

int state = 1;

int flag = 0; // make sure that you return the state only once

char d = 0;

char OP1[10];

char OP2[10];

char oprtr;

int opflag = 1;

int sizeofop1 = -1;

int sizeofop2 = -1;

int i = 0, j;

int num1 = 0, num2 = 0, temp, poww, result;

void setup() {

// sets the pins as outputs:

pinMode(ledPin, OUTPUT);

digitalWrite(ledPin, LOW);

Serial.begin(115200); // Default connection rate for my BT module

//Serial.println("Enter the input to increment : ");

}

void readInput(){

}

int power(int x, int y)

{

int temp =1, k;

for(k=0;k<y;k++)

temp = temp \* x;

return temp;

}

void loop() {

//if some data is sent, read it and save it in the state variable

if(Serial.available()){

d = Serial.read();

if(d=='+' || d=='-' || d=='\*' || d=='/')

{

oprtr = d;

// Serial.println((char)d);

opflag=0;

// opflag =1 means first operator is going on. store in OP1. else: OP2

Serial.println(" ");

Serial.print("Received op1 ");

Serial.println(OP1);

Serial.print("Received operator ");

Serial.println(oprtr);

}

else if(d=='x')

{

memset(OP1, 0, 10);

memset(OP2, 0, 10);

opflag =1;

sizeofop1=-1;

sizeofop2=-1;

num1 = 0;

num2 = 0;

Serial.println("Flushed everything");

Serial.println(OP1);

Serial.println(OP2);

Serial.println("Done.");

}

else if(d=='=')

{

Serial.print("Received op2 ");

Serial.println(OP2);

Serial.print("Parsed values: ");

Serial.print(OP1);

Serial.print(" and ");

Serial.println(OP2);

Serial.print("Result: ");

// compute op1 and op2

temp = sizeofop1;

for(j=0;j<=sizeofop1;j++)

{

poww = (int)power(10,temp);

num1 = num1 + (OP1[j]-48)\*poww;

temp--;

}

temp = sizeofop2;

for(j=0;j<=sizeofop2;j++)

{

poww = (int)power(10,temp);

num2 = num2 + (OP2[j]-48)\*poww;

temp--;

}

if(oprtr=='+')

result = num1 + num2;

else if(oprtr=='-')

result = num1 - num2;

else if(oprtr=='\*')

result = num1 \* num2;

else if(oprtr=='/' && num2)

result = num1 / num2;

Serial.println(result);

memset(OP1, 0, 10);

memset(OP2, 0, 10);

opflag =1;

sizeofop1=-1;

sizeofop2=-1;

num1 = 0;

num2 = 0;

}

else if(opflag)

{

sizeofop1++;

OP1[sizeofop1] = d;

}

else // opflag = 0

{

sizeofop2++;

OP2[sizeofop2] = d;

}

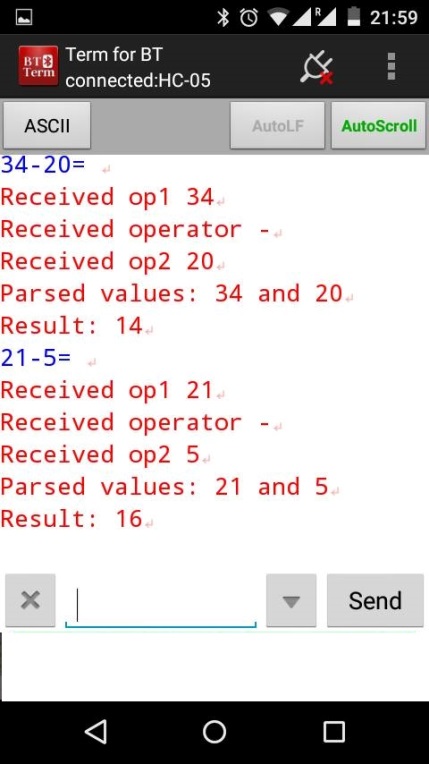
//delay(3000);

}

}

**Description** : In this code user has to input two inputs and an operator.According to the operator specified by the user the computation is done and the result is displayed at the S2 terminal and the serial communication interface.

**Screenshots:**



**FOURTH CODE:**

int ledPin = 13; // use the built in LED on pin 13 of the Uno

int state = 1;

int flag = 0; // make sure that you return the state only once

char d = 0;

char OP1[10];

char OP2[10];

char oprtr;

int opflag = 1, resultflag = 0;

int sizeofop1 = -1;

int sizeofop2 = -1;

int i = 0, j;

int num1 = 0, num2 = 0, temp, poww, result;

void setup() {

// sets the pins as outputs:

pinMode(ledPin, OUTPUT);

digitalWrite(ledPin, LOW);

Serial.begin(115200); // Default connection rate for my BT module

//Serial.println("Enter the input to increment : ");

}

void readInput(){

}

int power(int x, int y)

{

int temp =1, k;

for(k=0;k<y;k++)

temp = temp \* x;

return temp;

}

void loop() {

//if some data is sent, read it and save it in the state variable

if(Serial.available()){

d = Serial.read();

if(d=='+' || d=='-' || d=='\*' || d=='/' || d=='%')

{

oprtr = d;

if(resultflag)

{

num1 = result;

}

// Serial.println((char)d);

opflag=0;

// opflag =1 means first operator is going on. store in OP1. else: OP2

// Serial.println("Received operator ");

// Serial.println(oprtr);

}

else if(d=='x')

{

memset(OP1, 0, 10);

memset(OP2, 0, 10);

opflag =1;

sizeofop1=-1;

sizeofop2=-1;

num1 = 0;

num2 = 0;

//Serial.println("Flushed everything");

//Serial.println(OP1);

//Serial.println(OP2);

//Serial.println("Done.");

}

else if(d=='=')

{

//Serial.println("Received values: ");

//Serial.print(OP1);

//Serial.print(" and ");

//Serial.println(OP2);

//Serial.print("Result: ");

// compute op1 and op2

if(!resultflag)

{

temp = sizeofop1;

for(j=0;j<=sizeofop1;j++)

{

poww = (int)power(10,temp);

num1 = num1 + (OP1[j]-48)\*poww;

temp--;

}

}

temp = sizeofop2;

for(j=0;j<=sizeofop2;j++)

{

poww = (int)power(10,temp);

num2 = num2 + (OP2[j]-48)\*poww;

temp--;

}

if(oprtr=='+')

result = num1 + num2;

else if(oprtr=='-')

result = num1 - num2;

else if(oprtr=='\*')

result = num1 \* num2;

else if(oprtr=='/' && num2)

result = num1 / num2;

else if(oprtr=='%' && num2)

result = num1 % num2;

resultflag = 1;

Serial.println(result);

memset(OP1, 0, 10);

memset(OP2, 0, 10);

opflag =1;

sizeofop1=-1;

sizeofop2=-1;

num1 = 0;

num2 = 0;

}

else if(opflag)

{

result =0;

resultflag=0;

sizeofop1++;

OP1[sizeofop1] = d;

//Serial.print("Received op1 ");

//Serial.println(OP1);

}

else // opflag = 0

{

sizeofop2++;

OP2[sizeofop2] = d;

//Serial.print("Received op2 ");

//Serial.println(OP2);

}

//delay(3000);

}

}

**Description** : This code is an extension to the previous code in which the result of the previous computation is stored in a buffer.If the user wants to use the previous result and perform any operation on it he is allowed to do so .The result of the computation is displayed at the S2 terminal as well as the serial communication interface.

**Screenshots:**



**Final Program :**

#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins

LiquidCrystal lcd(8,9,4,5,6,7);

int ledPin = 13; // use the built in LED on pin 13 of the Uno

int state = 1;

int flag = 0; // make sure that you return the state only once

char d = 0;

char OP1[10];

char OP2[10];

char oprtr;

int opflag = 1, resultflag = 0;

int sizeofop1 = -1;

int sizeofop2 = -1;

int i = 0, j;

int num1 = 0, num2 = 0, temp, poww, result;

int a = 2,b=3,mm;

void setup() {

lcd.begin(16,2);

lcd.clear();

for(mm=0;mm<8;mm++){

lcd.setCursor(mm,0);

lcd.print('.');

lcd.setCursor(15-mm,0);

lcd.print('.');

delay(700);

lcd.clear();

}

lcd.clear();

lcd.setCursor(5,0);

lcd.print("CSE322");

lcd.setCursor(0,1);

lcd.print("Embedded Systems");

delay(2000);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Dr. Rajesh Kumar");

delay(1000);

lcd.clear();

lcd.setCursor(1,0);

lcd.print("Welcome to our");

lcd.setCursor(5,1);

lcd.print("Project");

delay(1000);

lcd.clear();

for(mm=0;mm<8;mm++){

lcd.setCursor(mm,0);

lcd.print('.');

lcd.setCursor(15-mm,0);

lcd.print('.');

delay(100);

lcd.clear();

}

lcd.clear();

lcd.setCursor(3,0);

lcd.print("Calculator");

// sets the pins as outputs:

pinMode(ledPin, OUTPUT);

digitalWrite(ledPin, LOW);

Serial.begin(115200); // Default connection rate for my BT module

//Serial.println("Enter the input to increment : ");

}

int power(int x, int y)

{

int temp =1, k;

for(k=0;k<y;k++)

temp = temp \* x;

return temp;

}

void loop() {

//if some data is sent, read it and save it in the state variable

if(Serial.available()){

d = Serial.read();

if(d=='+' || d=='-' || d=='\*' || d=='/' || d=='%')

{

oprtr = d;

if(resultflag)

{

num1 = result;

}

// Serial.println((char)d);

opflag=0;

// opflag =1 means first operator is going on. store in OP1. else: OP2

// Serial.println("Received operator ");

// Serial.println(oprtr);

}

else if(d=='x')

{

memset(OP1, 0, 10);

memset(OP2, 0, 10);

opflag =1;

sizeofop1=-1;

sizeofop2=-1;

num1 = 0;

num2 = 0;

//Serial.println("Flushed everything");

//Serial.println(OP1);

//Serial.println(OP2);

//Serial.println("Done.");

}

else if(d=='=')

{

//Serial.println("Received values: ");

//Serial.print(OP1);

//Serial.print(" and ");

//Serial.println(OP2);

//Serial.print("Result: ");

// compute op1 and op2

if(!resultflag)

{

temp = sizeofop1;

for(j=0;j<=sizeofop1;j++)

{

poww = (int)power(10,temp);

num1 = num1 + (OP1[j]-48)\*poww;

temp--;

}

}

temp = sizeofop2;

for(j=0;j<=sizeofop2;j++)

{

poww = (int)power(10,temp);

num2 = num2 + (OP2[j]-48)\*poww;

temp--;

}

if(oprtr=='+')

result = num1 + num2;

else if(oprtr=='-')

result = num1 - num2;

else if(oprtr=='\*')

result = num1 \* num2;

else if(oprtr=='/' && num2)

result = num1 / num2;

else if(oprtr=='%' && num2)

result = num1 % num2;

resultflag = 1;

Serial.println(result);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Result:");

lcd.setCursor(0,1);

lcd.print(result);

memset(OP1, 0, 10);

memset(OP2, 0, 10);

opflag =1;

sizeofop1=-1;

sizeofop2=-1;

num1 = 0;

num2 = 0;

}

else if(opflag)

{

result =0;

resultflag=0;

sizeofop1++;

OP1[sizeofop1] = d;

//Serial.print("Received op1 ");

//Serial.println(OP1);

}

else // opflag = 0

{

sizeofop2++;

OP2[sizeofop2] = d;

//Serial.print("Received op2 ");

//Serial.println(OP2);

}

//delay(3000);

}

}