

## Program Arduino

```
#include <EasyVR.h>

#include "Arduino.h"
#if !defined(SERIAL_PORT_MONITOR)

#if defined(__SAMD21G18A__)
// Shield Jumper on HW (for Zero, use Programming
Port)
#define port SERIAL_PORT_HARDWARE
#define pcSerial SERIAL_PORT_MONITOR
#elif defined(SERIAL_PORT_USBVIRTUAL)
// Shield Jumper on HW (for Leonardo and Due, use
Native Port)
#define port SERIAL_PORT_HARDWARE
#define pcSerial SERIAL_PORT_USBVIRTUAL
#else
// Shield Jumper on SW (using pins 12/13 or 8/9 as
RX/TX)
#include "SoftwareSerial.h"
SoftwareSerialport(12, 13);
#define pcSerial SERIAL_PORT_MONITOR
#endif

#include "EasyVR.h"

EasyVReasyvr(port);

//Groups and Commands
enum Groups
{
GROUP_0 = 0,
GROUP_1 = 1,
};

enum Group0
{
GO_START = 0,
};

enum Group1
{
G1_GO = 0,
G1_BACK = 1,
G1_STOP = 2,
G1_MAJU = 3,
G1_MUNDUR = 4,
G1_GOO = 5,
G1_BACKK = 6,
G1_STOPP = 7,
G1_GOOO = 8,
G1_BACKKK = 9,
G1_STOPPPP = 10,
G1_GOOOO = 11,
G1_BACKKKKK = 12,
G1_STOPPPPP = 13,
G1_GOOOOO = 14,
G1_BACKKKKKK = 15,
G1_STOPPPPPP = 16,
};

//EasyVRBridgebridge;

int8_t group, id x;

//motor in isialisasi
#define en1 6
#define in1 4
#define in2 2

//pwm 3,5,6,9,10,11
#define en2 3
#define in3 5
#define in4 7

#define t_lambat 19
#define t_sedang 18
#define t_cepat 17

#define lampu_merah 10
#define lampu_hijau 11

intsensor_lambat=0;
intsensor_sedang=0;
intsensor_cepat=0;

int a=0;
int b=0;
int status1=1;
voidmundur(){
//motor kiri
analogWrite(en1,a);
digitalWrite(in1,LOW);
digitalWrite(in2,HIGH);

//motor kanan
analogWrite(en2,b);
digitalWrite(in3,LOW);
digitalWrite(in4,HIGH);
}

voidmaju(){
analogWrite(en1,a);
digitalWrite(in1,HIGH);
digitalWrite(in2,LOW);

analogWrite(en2,b);
digitalWrite(in3,HIGH);
digitalWrite(in4,LOW);
}

voidberhenti(){
analogWrite(en1,0);
digitalWrite(in1,LOW);
digitalWrite(in2,LOW);

analogWrite(en2,0);
digitalWrite(in3,LOW);
digitalWrite(in4,LOW);
}

void setup()
{
//motor
pinMode (en1,OUTPUT); // jalur 5
pinMode (in1,OUTPUT);
```

```

pinMode (in2,OUTPUT);
pinMode (en2,OUTPUT);
pinMode (in3,OUTPUT);
pinMode (in4,OUTPUT);

//mode kecepatan
pinMode (t_lambat,INPUT);
pinMode (t_sedang,INPUT);
pinMode (t_cepat,INPUT);

// digitalWrite (t_lambat,HIGH);
// digitalWrite (t_sedang,HIGH);
// digitalWrite (t_cepat,HIGH);
//indikatorlampu
pinMode (lampu_merah,OUTPUT);
pinMode (lampu_hijau,OUTPUT);

// setup PC serial port
pcSerial.begin(9600);
bridge:
// bridge mode?
int mode = easyvr.bridgeRequested(pcSerial);
switch (mode)
{
caseEasyVR::BRIDGE_NONE:
// setup EasyVR serial port
port.begin(9600);
// run normally
pcSerial.println(F("Bridge not requested, run
normally"));
pcSerial.println(F(" ---"));
break;

caseEasyVR::BRIDGE_NORMAL:
// setup EasyVR serial port (low speed)
port.begin(9600);
// soft-connect the two serial ports (PC and Easy VR)
easyvr.bridgeLoop(pcSerial);
// resume normally if aborted
pcSerial.println(F("Bridge connection aborted"));
pcSerial.println(F(" ---"));
break;

caseEasyVR::BRIDGE_BOOT:
// setup EasyVR serial port (high speed)
port.begin(115200);
pcSerial.end();
pcSerial.begin(115200);
// soft-connect the two serial ports (PC and Easy VR)
easyvr.bridgeLoop(pcSerial);
// resume normally if aborted
pcSerial.println(F("Bridge connection aborted"));
pcSerial.println(F(" ---"));
break;
}

// initialize Easy VR
while (!easyvr.detect())
{
pcSerial.println(F(" EasyVR not detected!"));
for (int i = 0; i < 10; ++i)
{
if (pcSerial.read() == '?')

```

```

goto bridge;
delay(100);
}
}

pcSerial.print(F("Easy VR detected, version "));
pcSerial.print(easyvr.getID());

if (easyvr.getID() <Easy VR::EASYVR3)
easyvr.setPinOutput(EasyVR::IO1, LOW); // Shield
2.0 LED off

if (easyvr.getID() <Easy VR::EASYVR)
pcSerial.print(F(" = VRbot module"));
else if (easyvr.getID() <Easy VR::EASYVR2)
pcSerial.print(F(" = Easy VR module"));
else if (easyvr.getID() <Easy VR::EASYVR3)
pcSerial.print(F(" = Easy VR 2 module"));
else
pcSerial.print(F(" = Easy VR 3 module"));
pcSerial.print(F(", FW Rev.));
pcSerial.println(easyvr.getID() & 7);

easyvr.setDelay(0); // speed-up replies

easyvr.setTimeout(5);
easyvr.setLanguage(0);

group = EasyVR::TRIGGER; //<-- start group
(customize)
}

void loop()
{
awal();
}

voidawal(){
digitalWrite(t_lambat,HIGH);
digitalWrite(t_sedang,HIGH);
digitalWrite(t_cepat,HIGH);
Serial.println("Pilih Mode KecepatanAnda");

sensor_lambat = digitalRead(t_lambat);
sensor_sedang = digitalRead(t_sedang);
sensor_cepat = digitalRead(t_cepat);
if(sensor_lambat==LOW)
{
Serial.println("Mode lambat");
a=105;b=75;baca_suara();
}
else if(sensor_sedang==LOW)
{
Serial.println("Mode sedang");
a=150;b=120;baca_suara();
}
else if(sensor_cepat==LOW)
{
Serial.println("Mode cepat");
a=255;b=225;baca_suara();
}
}

```

```

}

void baca_suara(){
if (easyvr.getID() < EasyVR::EASYVR3)
easyvr.setPinOutput(EasyVR::IO1, HIGH); // LED on
(listening)
if (group < 0) // SI wordset/grammar
{
pcSerial.print("Say a word in Wordset ");
pcSerial.println(-group);
easyvr.recognizeWord(-group);
}
else // SD group
{
pcSerial.print("Say a command in Group ");
group=1;
pcSerial.println(group);
easyvr.recognizeCommand(group);
}
do
{
if (pcSerial.read() == '?')
{
setup();
return;
}
}
while (!easyvr.hasFinished());

if (easyvr.getID() < EasyVR::EASYVR3)
easyvr.setPinOutput(EasyVR::IO1, LOW); // LED off
idx = easyvr.getWord();
if (idx == 0 && group == EasyVR::TRIGGER)
{
// beep
easyvr.playSound(0, EasyVR::VOL_FULL);
pcSerial.println("Word: ROBOT");
return;
}
else if (idx >= 0)
{
// beep
easyvr.playSound(0, EasyVR::VOL_FULL);
// print debug message
uint8_t flags = 0, num = 0;
char name[32];
pcSerial.print("Word: ");
pcSerial.print(idx);
if (easyvr.dumpGrammar(-group, flags, num))
{
for (uint8_t pos = 0; pos < num; ++pos)
{
if (!easyvr.getNextWordLabel(name))
break;
if (pos != idx)
continue;
pcSerial.print(F(" "));
pcSerial.println(name);
break;
}
}
}
action();
return;

```

```

}
idx = easyvr.getCommand();
if (idx >= 0)
{
easyvr.playSound(0, EasyVR::VOL_FULL);
uint8_t train = 0;
char name[32];
pcSerial.print("Command: ");
pcSerial.print(idx);
if (easyvr.dumpCommand(group, idx, name, train))
{pcSerial.print(" ");
pcSerial.println(name);}
else
pcSerial.println();
action();
}
else // errors or timeout
{
if (easyvr.isTimeout())
pcSerial.println("Timed out, try again...");
int16_t err = easyvr.getError();
if (err >= 0)
{pcSerial.print("Error ");
pcSerial.println(err, HEX);
digitalWrite(lampu_merah, HIGH);}
}
baca_suara();
}

void action()
{
switch (group)

case GROUP_1:
switch (idx)
{
case G1_GO:
digitalWrite(lampu_merah, LOW);
digitalWrite(lampu_hijau, HIGH);
delay(500);
digitalWrite(lampu_hijau, LOW);
pcSerial.println("Maju");
maju();
delay(2000);
berhenti();
delay(500);
break;

case G1_BACK:
digitalWrite(lampu_merah, LOW);
digitalWrite(lampu_hijau, HIGH);
delay(500);
digitalWrite(lampu_hijau, LOW);
pcSerial.println("Mundur");
mundur();
delay(2000);
berhenti();
delay(500);
break;

case G1_STOP:
{digitalWrite(lampu_merah, HIGH);
digitalWrite(lampu_hijau, HIGH);

```

```
delay(500);
  // digitalWrite(lampu_hijau,LOW);
pcSerial.println("Stop");
awal();}
```

```
  // status1=0;
  // break;
```

```
case G1_MAJU:
digitalWrite(lampu_merah,LOW);
digitalWrite(lampu_hijau,HIGH);
delay(500);
digitalWrite(lampu_hijau,LOW);
pcSerial.println("Maju");
maju();
delay(2000);
berhenti();
```

```
delay(500);
break;
```

```
case G1_MUNDUR:
digitalWrite(lampu_merah,LOW);
digitalWrite(lampu_hijau,HIGH);
delay(500);
digitalWrite(lampu_hijau,LOW);
pcSerial.println("Mundur");
mundur();
delay(2000);
berhenti();
delay(500);
break;
```

```
  }
}
```

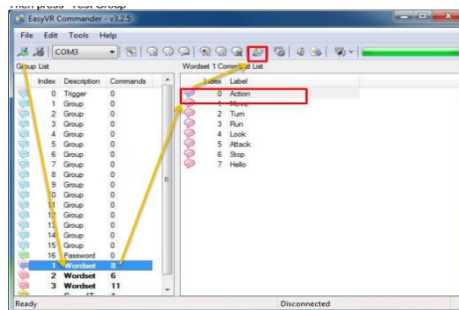
Dalam pengujian modul ini diperlukan PC dengan sistem operasi windows 7, sistem minimum dengan komunikasi serial seperti Arduino Uno. Langkah-langkah yang dilakukan sebagai berikut :

1. Menginstal *driver* sistem minimum pada komputer dan mengecek *device* tersebut sehingga dikenali sebagai COM Port pada *Device Manager*.
2. Menghubungkan jalur Rx, Tx, Vcc dan Ground pada modul *EasyVR* ke Arduino.
3. Membuka program *Easy Commander* dan mengatur nama COM Port pada *Combobox*.



Gambar 4.1 Pengaturan Nama Com

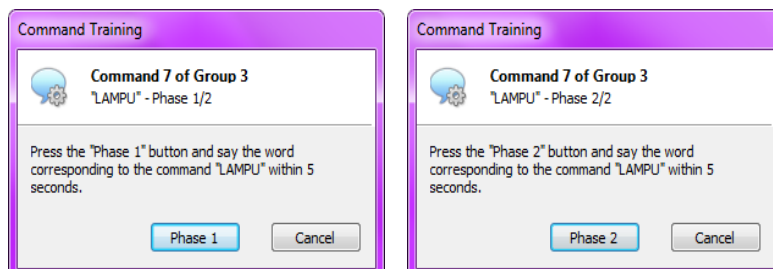
4. Kemudian memilih  *“Connect”*.  
 .Jika terhubung dengan benar maka akan muncul seperti tampilan dibawah ini:



Gambar 4.2 Tampilan *Device* Terhubung

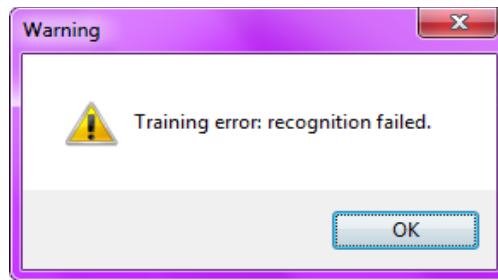
Setelah pengujian modul *EasyVR* berhasil, langkah selanjutnya melakukan perekaman bunyi suara sebagai pengendali robot. Langkah-langkah yang dilakukan sebagai berikut :

1. Memilih group pada *description* dalam hal ini terdapat 15 group dimulai dari index 1 sampai 15, index yang digunakan yaitu index 1.
2. Terdapat dua fase yang terjadi setiap kali perekaman bunyi suara pada masing-masing group, fase pertama dan kedua menandakan bahwa bunyi yang direkam di *training* sehingga dapat tersimpan dalam modul suara *EasyVR*.



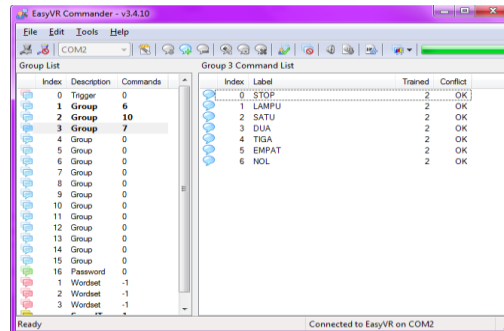
Gambar 4.3 Fase Perekaman Bunyi Suara

Jika bunyi suara tersebut belum tersimpan dalam modul *EasyVR*, maka akan muncul pesan sebagai berikut :



Gambar 4.4 TampilanPerekamanGagal

3. *Command* berjumlah 3 jenis dengan masing- masing label berupa *Go, Stop, Back.*



Gambar 4.5 Tampilan Group pada index ke 3