



EasyVR – SI/SD Commands and functions

Application Note

Release 2.0

```
void action()
{
    switch (group)
    {
        case GROUP_1:
            switch (idx)
            {
                case G1_REDLLED:
                    // write your action code here
                    // group = GROUP_X; <-- or jump to another group X for cc
                    break;
                case G1_GREENLED:
                    // write your action code here
                    // group = GROUP_X; <-- or jump to another group X for cc
                    break;
                case G1_RELAY:
                    // write your action code here
                    // group = GROUP_X; <-- or jump to another group X for cc
                    break;
            }
        break;
    }
}
```

Introduction

This Application Note shall teach how the common workflow goes with EasyVR Shields and Modules.

Please note this document does not replace the EasyVR User Manual, it just goes through the standard Speaker Dependent (SD) command creation process, up to the implementation of some actions that the microcontroller performs when a spoken command gets recognized.

This is the simplest example, the action performable are limited only by the specific Arduino board and your imagination. Whatever you plan to do in response to the spoken commands, start with switching a LED first. In this example we have 3 commands that switch 3 individual digital output pins.

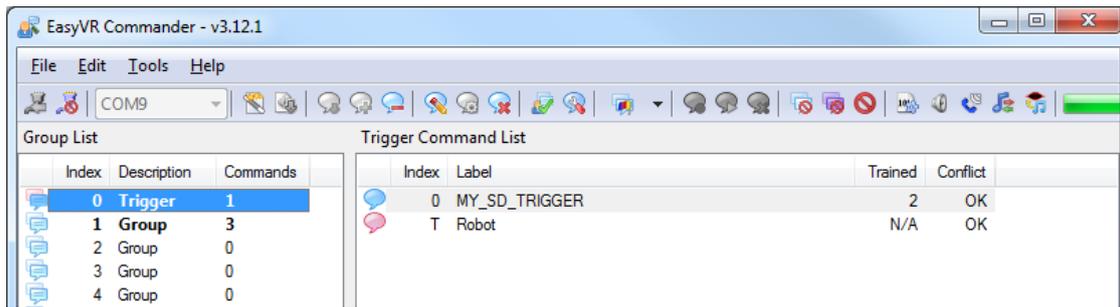
At the end of this document you will also find some notes about Arduino code for managing built-in or custom Speaker Independent (SI) grammars. The source code of this example is available in the same ZIP archive containing this document.

This Application Note refers to EasyVR Commander starting from version 3.12.1 and EasyVR Arduino library starting from version 1.10.1 and it may or may NOT work with previous versions.

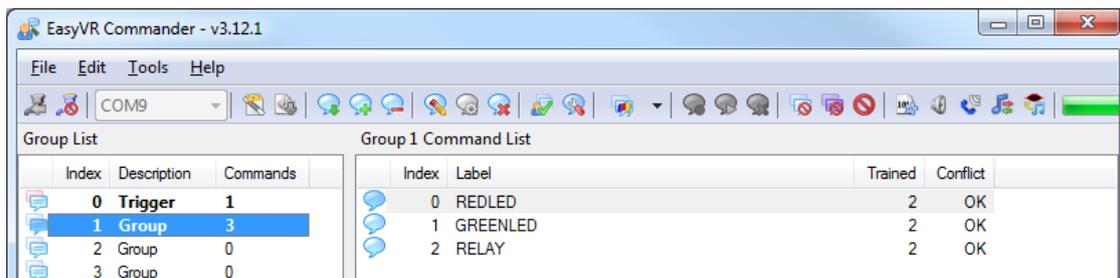
Before going on please carefully read the paragraph *Quick start guide for using the Shield* on the EasyVR 3 User Manual ([available here](#)).

Have fun,
Your VeeAR Support Team

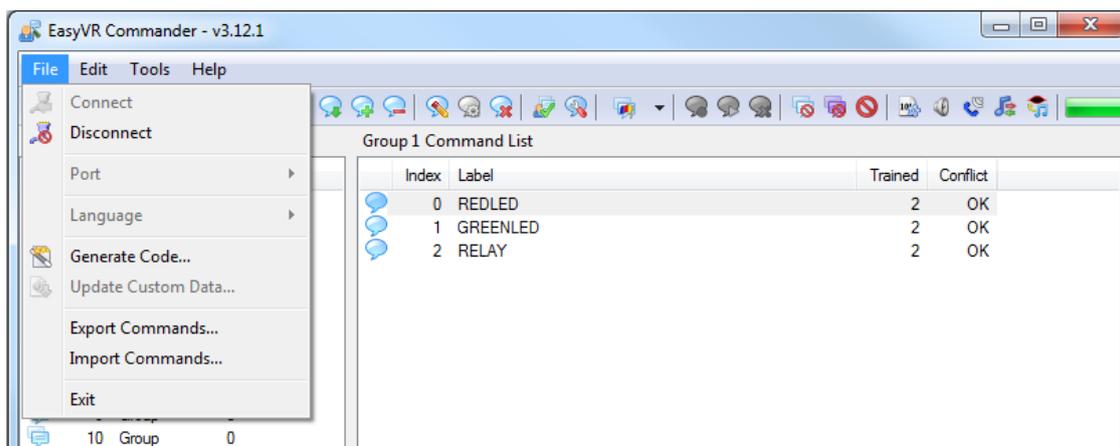
1. Check for the built-in Trigger Word “Robot” and add a custom speaker dependent (SD) trigger word (MY_SD_TRIGGER in this example):



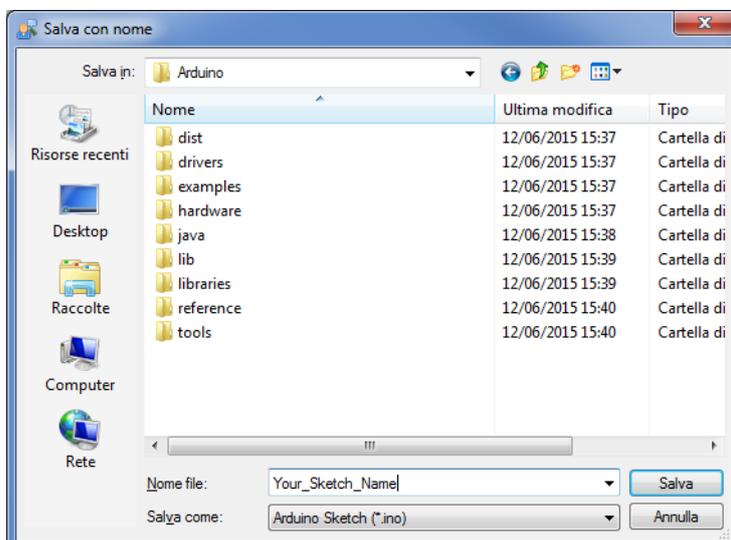
2. Create and train three Commands in Group 1, as in the following example:



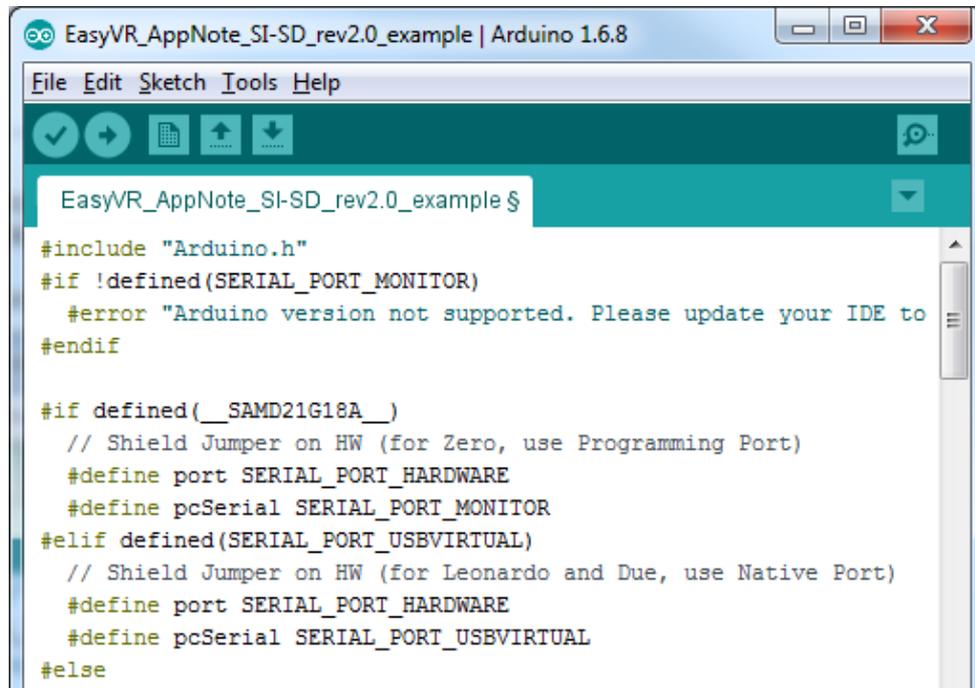
3. Get the code for Arduino from the EasyVR Commander (File->Generate Code...):



4. Save the code as Arduino Sketch (*.ino):



5. Go to the folder where you exported the code to and double click the file, then click on Yes when Arduino alerts you that a new folder with the same name of your sketch will be created.
6. The exported code will open in Arduino IDE:



7. Have a look at the last line of the `setup()` function, this is the group where recognition starts (“EasyVR::TRIGGER” i.e. Group 0, in this case):

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

8. In this example we will switch two LEDs and one relay.

Add the following code in void setup:

```
void setup()
{
  pinMode (4, OUTPUT); //red LED
  pinMode (5, OUTPUT); //green LED
  pinMode (6, OUTPUT); //relay
```

9. Add the following code in yellow to jump to GROUP_1 when the built-in SI trigger word “ROBOT” is recognized:

```
idx = easyvr.getWord();
if (idx == 0 && group == EasyVR::TRIGGER)
{
  // beep
  easyvr.playSound(0, EasyVR::VOL_FULL);
  // print debug message
  pcSerial.println("Word: ROBOT");
  // write your action code here
  group = GROUP_1;
  // group = GROUP_X\SET_X; <-- jump to another group or wordset
  return;
}
```

10. Scroll down where you find the following code:

```
case GROUP_1:
  switch (idx)
  {
    case G1_REDLED:
      // write your action code here
      // group = GROUP_X\SET_X; <-- or jump to another group or
wordset for composite commands
      break;
    case G1_GREENLED:
      // write your action code here
      // group = GROUP_X\SET_X; <-- or jump to another group or
wordset for composite commands
      break;
    case G1_RELAY:
      // write your action code here
      // group = GROUP_X\SET_X; <-- or jump to another group or
wordset for composite commands
      break;
  }
```

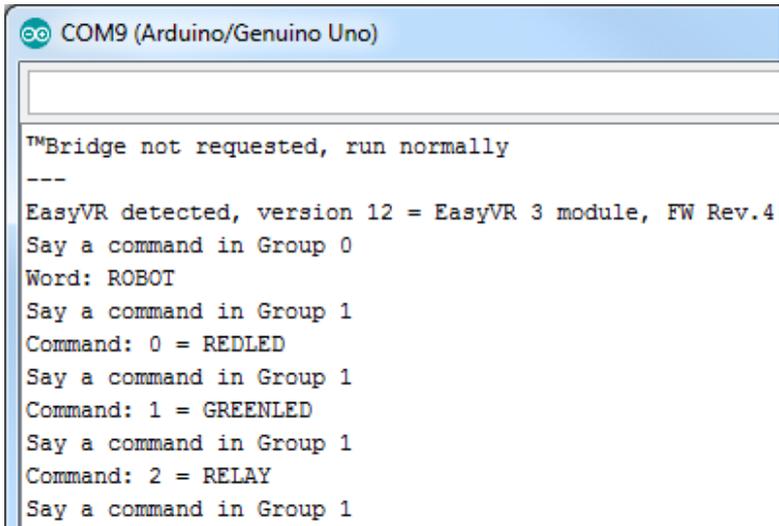
11. Add some action code here:

```
case GROUP_1:
  switch (idx)
  {
    case G1_REDLED:
      // write your action code here
      digitalWrite(4, HIGH);
      // group = GROUP_X\SET_X; <-- or jump to another group or
wordset for composite commands
      break;
    case G1_GREENLED:
      // write your action code here
      digitalWrite(5, HIGH);
      // group = GROUP_X\SET_X; <-- or jump to another group or
wordset for composite commands
      break;
    case G1_RELAY:
      // write your action code here
      digitalWrite(6, HIGH);
      // group = GROUP_X\SET_X; <-- or jump to another group or
wordset for composite commands
      break;
  }
```

12. Remember to disconnect the EasyVR Commander and then upload your sketch to the Arduino board.

Let's recap what this sketch will do: it starts to recognize in the trigger group (i.e. Group 0) and, if it recognizes the built-in trigger word "ROBOT", it jumps to Group 1 (where you previously trained three SD commands: REDLED, GREENLED and RELAY). Then it switches one of the output pins high depending on the recognized word. It will remain indefinitely in Group 1 and you have to reset Arduino to start from the beginning in Group 0.

13. You can start the serial monitor to see what is going on:



```
COM9 (Arduino/Genuino Uno)
Bridge not requested, run normally
---
EasyVR detected, version 12 = EasyVR 3 module, FW Rev.4
Say a command in Group 0
Word: ROBOT
Say a command in Group 1
Command: 0 = REDLED
Say a command in Group 1
Command: 1 = GREENLED
Say a command in Group 1
Command: 2 = RELAY
Say a command in Group 1
```

Additional Tips&Tricks

Jumping from one set to another

The above example remains indefinitely in Group 1 after the built-in SI trigger word “ROBOT” is recognized. If you want to jump back to Group 0 once a command in Group 1 is recognized, you just need to add the following code (note that *Group_0* and *EasyVR::TRIGGER* are both equal to 0):

```
case GROUP_1:
  switch (idx)
  {
    case G1_REDLLED:
      // write your action code here
      digitalWrite(4, HIGH);
      group = GROUP_0;
      // group = GROUP_X\SET_X; <-- or jump to another group or
wordset for composite commands
      break;
    case G1_GREENLED:
      // write your action code here
      digitalWrite(5, HIGH);
      group = GROUP_0;
      // group = GROUP_X\SET_X; <-- or jump to another group or
wordset for composite commands
      break;
    case G1_RELAY:
      // write your action code here
      digitalWrite(6, HIGH);
      group = GROUP_0;
      // group = GROUP_X\SET_X; <-- or jump to another group or
wordset for composite commands
      break;
  }
}
```

Custom SD trigger

The command `easyvr.getWord()` in the following part of the code is used to get the built-in trigger word “ROBOT”:

```
idx = easyvr.getWord();
if (idx == 0 && group == EasyVR::TRIGGER)
{
  // beep
  easyvr.playSound(0, EasyVR::VOL_FULL);
  // print debug message
  pcSerial.println("Word: ROBOT");
  // write your action code here
  group = GROUP_1;
  // group = GROUP_X\SET_X; <-- jump to another group or wordset
  return;
}
```

If you train a custom Speaker Dependent trigger word in Group 0 (MY_SD_TRIGGER in this example), the code generated by the EasyVR Commander will include it as in the following example:

```
void action()
{
  switch (group)
  {
    case GROUP_0:
      switch (idx)
      {
        case G0_MY_SD_TRIGGER:
          // write your action code here
          // group = GROUP_X\SET_X; <-- or jump to another group or
wordset for composite commands
          break;
        }
      break;
    case GROUP_1:
      switch (idx)
      {
        case G1_REDLIED:
          // write your action code here
          digitalWrite(4, HIGH);
          group = GROUP_0;
          // group = GROUP_X\SET_X; <-- or jump to another group or
wordset for composite commands
          break;
      }
    }
  }
}
```

If you want to jump to Group 1 also when the custom SD trigger is recognized from Group 0, you just need to modify the code as follows:

```
case G0_MY_SD_TRIGGER:
  // write your action code here
  group = GROUP_1;
  // group = GROUP_X\SET_X; <-- or jump to another group or
wordset for composite commands
  break;
```

This way, also when the custom SD trigger is recognized from Group 0, it will jump to Group 1 (thanks to the instruction “*group = GROUP_1;*”)

The sketch “*EasyVR_AppNote_SI-SD_rev2.0_example1.ino*” includes all the above code.

Using SI commands and custom Grammars

Using built-in SI commands or custom grammars follows exactly the same approach of using SD commands, the only difference is that you have to use `easyvr.recognizeWord(n)` instead of `easyvr.recognizeCommand(n)` and then `easyvr.getWord()` instead of `easyvr.getCommand()` to retrieve the recognized command index.

Starting from EasyVR Commander version 3.12.1, the generated code helps you to easily manage SI built-in commands or custom grammars. In fact, it automatically enumerates all built-in and available custom grammars at the beginning of the code, so that you can use them with their labels instead of numbers.

For instance, the first built-in set is identified as `SET_1` and the first command in this set is identified as `S1_ACTION`.

We created a simple example code "`EasyVR_AppNote_SI-SD_rev2.0_example2.ino`" and you can find it attached to this application note.

The example assumes you downloaded the freely available custom grammar `HomeAutomation_1` to your module and starts waiting for the custom SI trigger word "`OK EasyVR`" (in wordset number 4). If the recognition is successful, it then jumps to wordset number 5.

After a command in wordset number 5 is recognized, it jumps back to wordset number 4 waiting for the trigger "`OK EasyVR`".

Of course you can customize the code by adding what you need in the `action()` function.

As usual, you can use the Arduino IDE "Serial monitor" to see what is happening while running the code.

How to get support

Manuals / Application Notes / Demo Code

A user manual which includes all the information required to get started is provided in the [download section](#). All of the necessary software is installed together with the EasyVR Commander. This includes QuickSynthesis and FluentChip and Quick T2SI Lite with all available Language packs. Please note that although the Quick T2SI Lite is installed with the EasyVR Commander, a user license is available separately to activate this part of the installation. The Quick T2SI Lite license enables creation of custom Speaker Independent Commands. Additional libraries and examples for Arduino are also available. Please check the [download section](#) for more details.

Contact

Please feel free to contact us with any questions, queries or suggestions. If your question is about technical support or troubleshooting for one of our products, we kindly ask you to first check the user manual for a possible solution. If you cannot find an existing solution in the available resources, please contact us on support@robotechsrl.com. The more detail you provide, the better support we can give.

VeeAR © RoboTech srl, all rights reserved.



ROBOTECH

All VeeAR branded boards and software are manufactured by RoboTech srl

RoboTech srl assumes no responsibility for any errors, which may appear in this manual. Furthermore, RoboTech srl reserves the right to alter the hardware, software, and/or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. RoboTech srl products are not authorized for use as critical components in life support devices or systems.