

---

# Sharing Data Between 2 Flexys

---

Doc Created By: Tim Beaulieu

Code Created by: Deryck Beliveau

Version 1.1

Created on: 6/11/2019

Last Updated: 6/12/2019

**Contents**

Requirements: .....2

Step 1: Creating Sharable Tags .....3

Step 2: BASIC IDE Code .....3

Step 3: Sending data .....4

Appendix: .....5

    Developer ID Link:.....5

    Full Code: .....5

## Requirements:

- Need to make sure that **each tag you want to push** has a tag with the **same name** on the **other device**
- Each Tag that you want to share needs to have the “Tag description” say **send**
- Need to make sure the devices have each **passed** the **Internet** and **VPN wizard**
- Need a **Developer ID** (see appendix)
- Make sure when you enter the device name that there’s no space. Use an underscore instead.  
EX: **Deryck\_Flexy** works but **Deryck Flexy** will confuse the code

## Step 1: Creating Sharable Tags

For this example, we're going to create a tag called test. The way that the BASIC IDE code is written is to make it so that if a tag has the description "send" then it will share data to the other flexy that has a tag matching the same name.

Name	Type	IO Server	Topic	IO Address	Value
test	Floating po...	MEM			5
analog_exa...	Floating po...	MEM	analog_exa...		0
check1	Floating po...	MEM	check1		0
tag2	Floating po...	MEM	tag2		0
modbus1	Floating po...	MEM	modbus1		0
readmodbus	Floating po...	MODBUS	A	40002,20,1...	0
stringpoint	Integer	MEM	stringpoint		0
num	Integer	MEM	num		0
initialize	Floating po...	MEM	initialize		0
Digital_Out...	Floating po...	EWON	DO1		1
tag1	Floating po...	MEM	tag1		0
Tag3	Floating po...	MEM	Tag3		0
delay_count	Floating po...	MEM	delay_count		0
start	Floating po...	MEM	start		0
runtime	String	MEM	runtime		
count	Integer	MEM	count		0
min	Integer	MEM	min		0
hour	Integer	MEM	hour		0
Vessel_Batch	Integer	MEM	Vessel_Batch		0
Vessel_Cycl...	Floating po...	MEM	Vessel_Cycl...		0

Figure 1: Showing what the first tag "test" needs to look like on each of the Flexys

## Step 2: BASIC IDE Code

This section of code is written so that it will have a timer go off every 5 seconds and then push the values to the other flexy in the update remote section. It also will have you enter the account information of the account for the Flexy that you're working with as well as the eCatcher username/password. The developer ID needed to run the code can be gotten from a link at the bottom of the document. The "devicename" is what you called the device that you're writing to in eCatcher and the username and password are the logins for that specific device.

```
TSET 1, 5
ONTIMER 1, "GOTO UpdateRemote"
ONSTATUS "@response()"
//Account info
account$ = "tims"
username$ = "script_user"
password$ = "script_user"
developerid$ = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
//info for remote flexy
devicename$ = "Deryck_Flexy"
deviceusername$ = "adm"
devicepassword$ = "adm"
url$ = "https://m2web.talk2m.com/t2mapi/get/"+devicename$+"/rog1.bin/UpdateTagForm?"
accountinfo$ = "&t2maccount="+account$+"&t2musername="+username$+"&t2mpassword="+password$
accountinfo$ = accountinfo$ + "&t2mdeveloperid="+developerid$+"&t2mdeviceusername="+deviceusername$ + devicepassword$
```

Figure 2: This is just showing the section of the code that needs to be edited to your specific parameters.

### Step 3: Sending data

At this point, all the information above should have been edited and you can copy and paste in the rest of the code from the appendix into the BASIC IDE. Next hit the Script execution button and you should now be able to push data to the other device.

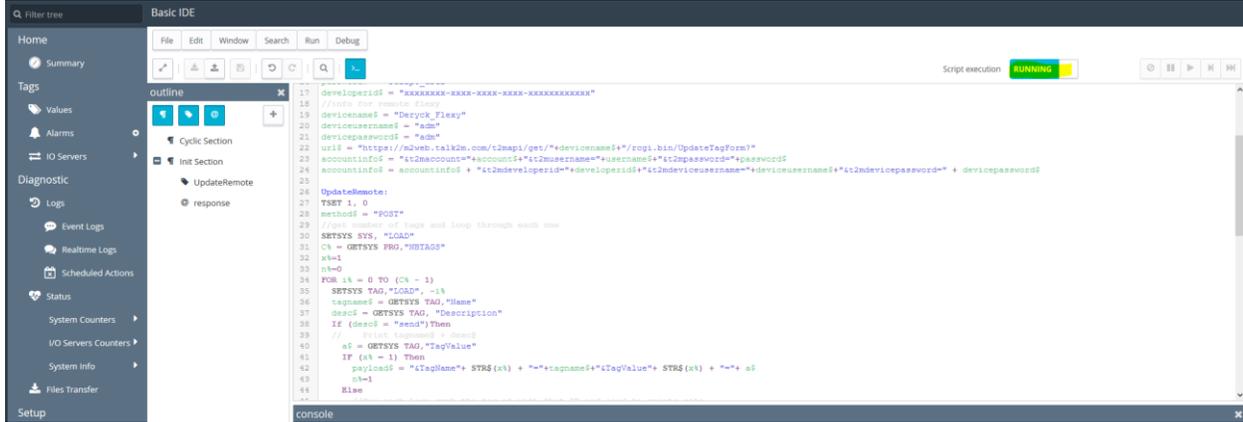


Figure 3: Showing what it should look like inside the BASIC IDE section

To check if this is working, go to the values section and make sure you're in the view mode. Now we should be able to click on the tag "test" and go to edit value. After making a change on the device that has the code running on it, you should notice the second device is showing the same value. It is essentially pushing the value from the flexy with the BASIC code onto the other device by writing to it over the M2Web connection.

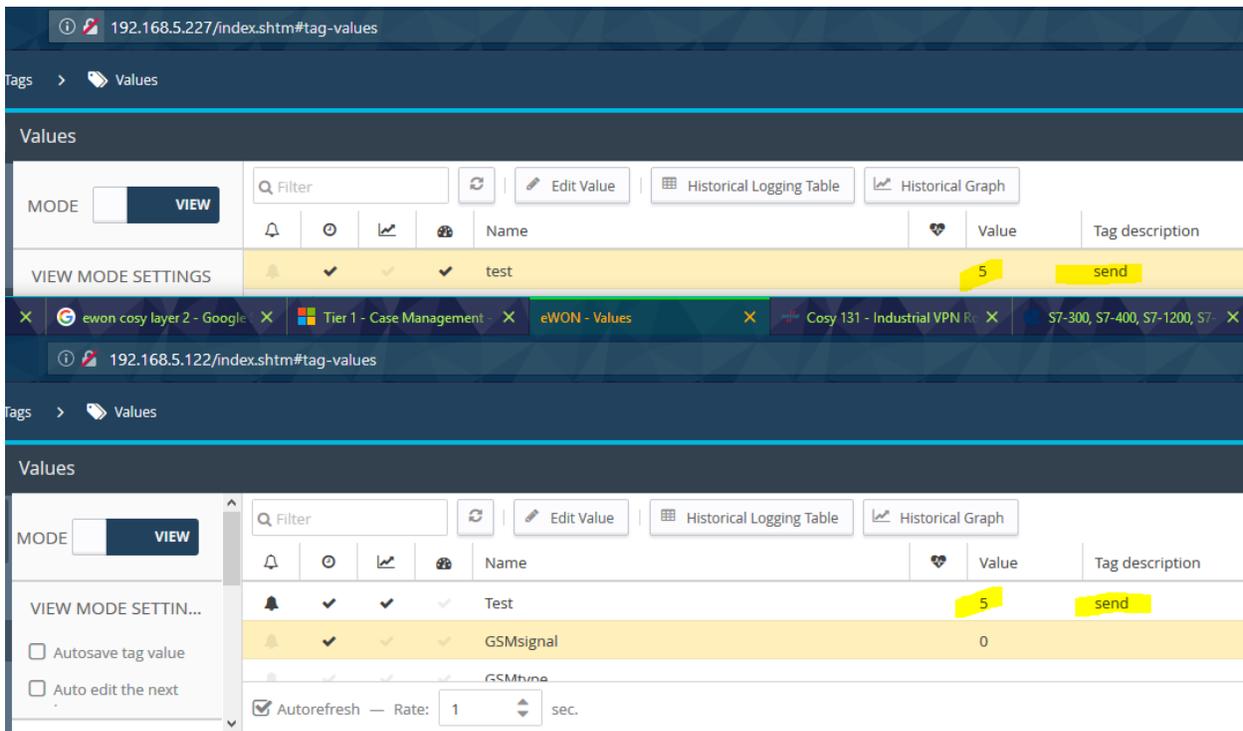


Figure 4: Example showing what it looks like on each flexy.

## Appendix:

### Developer ID Link:

<https://developer.ewon.biz/content/talk2m-developer-id>

### Full Code:

```
TSET 1, 5

ONTIMER 1, "GOTO UpdateRemote"

ONSTATUS "@response()"

//Account info

account$ = "tims"

username$ = "script_user"

password$ = "script_user"

developerid$ = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"

//info for remote flexy

devicename$ = "Deryck_Flexy"

deviceusername$ = "adm"

devicepassword$ = "adm"

url$ = "https://m2web.talk2m.com/t2mapi/get/" + devicename$ + "/rcgi.bin/UpdateTagForm?"

accountinfo$ = "&t2macount="+account$+"&t2musername="+username$+"&t2mpassword="+password$

accountinfo$ = accountinfo$ + "&t2mdeveloperid="+developerid$+"&t2mdeviceusername="+deviceusername$+"&t2mdevicepassword="+devicepassword$

UpdateRemote:

TSET 1, 0

method$ = "POST"

//get number of tags and loop through each one

SETSYS SYS, "LOAD"

C% = GETSYS PRG, "NBTAGS"

x%=1

n%=0

FOR i% = 0 TO (C% - 1)

  SETSYS TAG, "LOAD", -i%

  tagname$ = GETSYS TAG, "Name"

  desc$ = GETSYS TAG, "Description"

  If (desc$ = "send")Then

    // Print tagname$ + desc$

    a$ = GETSYS TAG, "TagValue"

    IF (x% = 1) Then

      payload$ = "&TagName"+ STR$(x%) + "=" + tagname$ + "&TagValue" + STR$(x%) + "=" + a$

      n%=1

    Else

      //for each loop grab the tag at with that ID and send to remote site
```

```

payload$ = payload$ + "&TagName" + STR$(x%) + "=" + tagName$ + "&TagValue" + STR$(x%) + "=" + a$
ENDIF

x%=x%+1

//Print "success "+"index: " + STR$(i%)

Endif

NEXT i%

//Print payload$

If (n% = 1)Then

payload$ = payload$ + accountinfo$

REQUESTHTTP url$, method$, "" , payload$

actionID% = GETSYS PRG, "ACTIONID"

//ONSTATUS "@response()"

Print "Data sent"

Endif

TSET 1, 5

END

FUNCTION response()

eventId% = GETSYS PRG, "EVTINFO"

IF (eventId% = actionID%) THEN

SETSYS PRG, "ACTIONID", eventId%

stat% = GETSYS PRG, "ACTIONSTAT"

IF (stat% = 0) THEN

b$ = RESPONSEHTTP "STATUSCODE"

IF (b$ <> "200") THEN

LOGEVENT "HTTP RESPONSE: " + b$, 77

//can add an event here on error

ENDIF

ELSE

LOGEVENT "Error (ERROR = "+Str$(stat%) + ")" , 78

//LOGEVENT url$, 79

//LOGEVENT payload$, 80

ENDIF

ENDIF

Endfn

```