



life.augmented

How to add a register to MC FW and Pilot

- 2 actions are needed:
 1. Modify Motor Control Firmware to add your new register
 2. Modify register description file in Motor Pilot

add definition of your new register in FW

- Add register definition to header register_interface.h
 - Example for 32 bits register

```
#define MC_REG_NEW ((112 << ELT_IDENTIFIER_POS) | TYPE_DATA_32BIT )
```



New
register Id



32 bits

add read/Write access of your new register in FW

- Add write register in `RI_SetReg` function (register_interface.c)
 - Example for 32 bits register

```
case TYPE_DATA_32BIT:
{
    uint32_t regdata32 = *(uint32_t *)data; //cstat !MISRAC2012-Rule-11.3

    switch (regID)
    {
        case MC_REG_NEW:
        {
            //do your stuff with regdata32
            break;
        }
    }
}
```



add read/Write access of your new register in FW

- Add read register in `RI_GetReg` function (register_interface.c)
 - Example for 32 bits register

```
case TYPE_DATA_32BIT:
{
    uint32_t *regdataU32 = (uint32_t *)data; //cstat !MISRAC2012-Rule-11.3
    int32_t *regdata32 = (int32_t *)data; //cstat !MISRAC2012-Rule-11.3

    if (freeSpace >= 4U)
    {
        switch (regID)
        {
            case MC_REG_NEW:
            {
                *regdataU32 = your_NEW_REG_value;
                break;
            }
        }
    }
}
```

add High frequency Plot capability to your new register in FW

- Add pointer in `RI_GetPtrReg` function (register_interface.c)
- Only 16 bit are sent to Pilot (if you have 32 bit register you must reduce it before send)
 - Example for 16 bits register

```
switch (typeID)
{
    case TYPE_DATA_16BIT:
    {
        switch (regID)
        {
            case MC_NEW_REG:
            {
                *dataPtr = &(NEW_REG_VALUE_16BITS);
                break;
            }
        }
    }
}
```



Add register into Motor Pilot

- The register description file is a json file loaded by Motor Pilot to populate the register list
- The register description file path is defined in the qml of your UI
 - Example for defaultApp.qml :
 - `property string registerFileDescription: "RegListSTMV2.json"`
- You can add your new Register to this file :

```
{  
  "Id" : 112,  
  "Name" : "REG_NEW",  
  "Description" : "new register description",  
  "Type" : "F32",  
  "Unit" : "new register unit",  
  "Access" : "RW",  
  "UpdateFrequency": "HighFrequency"  
}
```

New register Id (same as Firmware)

Type of New register Id (same as Firmware)

If you want to plot with High Frequency plot

- Qml file must be reload after modification in json file

