CFG

- General
- SYS.CFG
- SIO.CFG

General

Topic	Parameters that affect	File name
Controller	Event routines, etc.	SYS.CFG
Communication	Serial channels	SIO.CFG
IO Signals	I/O boards and signals	EIO.CFG
Manipulator	The robot and external axes	MOC.CFG
Teach Pendant	Displaying data and access on the teach pendant	MMC.CFG
Arc Welding	Arc welding	PROC.CFG

Estos archivos vienen definidos por defecto en el disquete de System Parameters.

No hay documentación oficial de aquella época que haya encontrado que diga cómo se estructura los archivos de configuración, así pues toda información aquí se ha extraído por comparación entre varios archivos y "lógica" que parece tener. Si bien es cierto, que hay pistas en documentación actual.

En todos estos archivos de configuración son archivos de texto plano y hay una cabecera al inicio del archivo del estilo:

SYS:CFG_1.0:2:0::

Estos parámetros vienen separados por : y siguiendo este ejemplo tenemos que,

SYS haciendo referencia al nombre del archivo,

CFG_1.0 haciendo referencia a que pueda haber más de una configuración

Seguido de 2:0:: esto para determinar a qué versión del software pertenece ya que también se ha encontrado archivos con 2:1::

Veáse el apartado de SYS.CFG.

Puede contener algo de información útil, pero no es directa, con pinzás.

https://manualsnet.com/abb/robotware-add-ins

Note

The cfg files must start with the name of the domain since the config command uses this information to determine the domain. The first row in the cfg file shall contain the following information where <version> and <revision> are optional:

```
<domain name>:CFG_1.0:<version>:<revision>::
```

Example:

```
EIO:CFG_1.0:: Domain EIO without version and revision
```

2 RobotWare Add-In functionality

2.3.2 Commands
Continued

Erase unprotected contents

The `config` command can be used to erase unprotected contents of a configuration domain before loading a new content.

Parameter	Description	Default
domain	The topic of the *cfg* file. Allowed values are: • *SIO* - Communication • *SYS* - Controller • *EIO* - I/O • *MMC* - Man-machine communication • *MOC* - Motion	
erase	Erase unprotected instances of the specified configuration domain.	FALSE

2.6 System parameters related to add-in development

2.6.1 About cfg files

Overview

The cfg files are used to define instances of system parameter types in a specific domain. The specified instances are then created by loading the cfg file. Only one domain can be specified per cfg file.

The file shall be formatted according to the rules in the following sections.

Domain specifier

A cfg file must start with a name of a domain where the specified instances will be created.

The row must contain the following information, where `<version>` and `<revision>` are optional:

```
<domain name>:CFG_1.0:<version>:<revision>::
```

Example

`EIO:CFG_1.0::`	Domain EIO without version number
`EIO:CFG_1.0:5:0::`	Domain EIO with version number 5.0
`EIO:CFG_1.0:6:0::`	Domain EIO with version number 6.0

2.6.2 Topic Controller

About the topic Controller

This section describes system parameters that belong to the topic *Controller* (that is, in the configuration file sys.cfg) and that are closely related to add-in development.

The configuration of which program modules to load is made in the topic *Controller*. All files containing the RAPID code for the add-in must be defined here.

For more information about the types and parameters of the *Controller* topic, see *Technical reference manual - System parameters*.

Automatic loading of modules (CAB_TASK_MODULES)

The type CAB_TASK_MODULES is used to define modules to be loaded when the controller is started.

For more information, see *Technical reference manual - System parameters*.

Parameter	Description
File	The name of the file including the path on the controller. An environment variable can preferably be used. That is, *<environment variable>:/<file name>*. See *setenv on page 29*.
Task	Name of a task, if it should only be loaded to one specific task. **Note** The parameters *Task*, *Shared*, *AllTask* and *AllMotionTask* are mutually exclusive.
Shared	Defines if the contents of a module should be reachable from all tasks. The module is not loaded, it is installed, but reachable from all tasks. **Note** The parameters *Task*, *Shared*, *AllTask* and *AllMotionTask* are mutually exclusive. **Note** The parameter *Shared* cannot be combined with *Installed*.
AllTask	Defines if the module should be loaded into all tasks. **Note** The parameters *Task*, *Shared*, *AllTask* and *AllMotionTask* are mutually exclusive.
AllMotionTask	Defines if the module should be loaded into all motion tasks. **Note** The parameters *Task*, *Shared*, *AllTask* and *AllMotionTask* are mutually exclusive.

Continues on next page

© Copyright 2015-2022 ABB. All rights reserved.

Comments

A comment row starts with '#'.

Type specifiers

The domain specifier is followed by one or more parameter type specifiers and their instances.

- A type specifier should always be preceded by a row containing a single character '#'. (Not mandatory)
- A type specifier consists of a parameter type name directly followed by a ':'.
- There should be an empty row between the type name and the first instance. (Not mandatory)
- There should be no more rows after the last instance row in a cfg file. (Not mandatory)
- Add a description of all attributes in a type directly after the type specifier. This is helpful for the user to understand the type. (Not mandatory)

See cfg file examples later in this section.

Instances and attributes

The type specifier is followed by zero or more instances. Each instance contains one or several attributes defining its properties. Attributes can be mandatory or optional.

Continues on next page

© Copyright 2015-2022 ABB. All rights reserved.

2.6.1 About cfg files

Continued

```
COM_TRP:
  # -Name Name of transmissions protocol (MAN)
  # -Type Name of transmissions protocol type (MAN)
  # -PhyChannel Name of the physical channel (MAN)
  # -HostName Name of host (OPT)
  # -RemoteAdress Remote address (OPT)
  # -Gateway Default gateway (OPT)
  # -SubnetMask SubNetmask (OPT)

  -Name "TCPIP1" -Type "TCP/IP" -PhyChannel "LAN1"
```

Example of cfg file

```
SIO:CFG_1.0::
#
COM_PHY_CHANNEL:

  -Name "COM1" -Connector "COM1"
  -Name "LAN1" -Connector "LAN"
#
```

Example

```
EIO:CFG_1.0::
#
EIO_SIGNAL:
-Name "DOAccessInternal" -SignalType "DO" -Access "internal"
-Name "DOAccessInternalRAPID" -SignalType "DO" -Access "internal"
     -Category "rapid"
```

SYS.CFG

Aquí tenemos dos ejemplos de configuración tanto para la vesión 2.0 cómo la versión 2.1, ambos comparten los mismos campos. Explicados en orden de aparición.

NOMBRE	**FUNCIÓN**
(*) SYS_MISC	??? (*) Sólo disponible en la vesión 2.1.
CAB_TASKS	Define las **tareas de ejecución** del robot en el controlador: • Nombre • Tipo (NORMAL, SEMISTATIC, etc.) • Programa asociado • Punto de entrada Esto controla qué procesos funcionan simultáneamente.
	Configuraciones de distancias y tolerancias para la recuperación de posición del robot, usadas en movimientos automáticos y manuales.
CAB_EXEC_HOOKS	Estos son **“ganchos” o hooks del sistema** que asocian programas o rutinas con eventos internos como: • POWER_ON • START • RESET Imagina que es un *trigger* que dice “cuando ocurra X, ejecutar Z”
CAB_TASK_MODULES	Este bloque especifica **qué módulos de código se cargan** para cada tarea definida: • Nombre del archivo (.sys, .mod, .prg) • Nombre del módulo • A qué tarea va asociado Esto es lo que realmente hace que el robot sepa qué código ejecutar para cada parte del sistema.
MASTER_BOOL	???
AUTH_INTERACTIVE_UNITS	???
AUTH_CLIENTS	???
AUTH_CLIENT_PERMISSION	???

WORKDIR	???
MMC_ATTRIBUTE	???
MMC_FKEY	???
MMC_OUTLINE	???
MMC_CHOICE	???

CAB_EXEC_HOOKS:

-Routine "RESTORE_IO" -Shelf "POWER_ON" -Task "0"

Vamos a desglosar aquí que ocurre, la rutina (función de programación) RESTORE_IO (en este caso, definida en el archivo base.sys) es lanzada cuando se produce el evento de POWER_ON.

La tarea 0 no sé si está definida en algún sitio o es una forma de referirse al sistema operativo o una tarea que se ejecuta siempre en por encima de todo, al estilo SO.

```
SYS:CFG_1.0:2:0::
#
#
#
CAB_TASKS:

    -Name "MAIN" -Prog 0 -Task_in_forground "" -Type "NORMAL" -Entry "main"\
    -RewERIfStop 0
#
CAB_REGAIN_DIST:

    -Name "MAN" -Tcp_dist 0.010000 -Tcp_rot 0.200000 -Ext_dist 0.050000\
    -Ext_rot 0.200000

    -Name "AUTO" -Tcp_dist 0.500000 -Tcp_rot 1.570000 -Ext_dist 0.500000\
    -Ext_rot 1.570000
#
```

```
CAB_EXEC_HOOKS:

    -Routine "SYS_RESET" -Shelf "RESET" -Task "0"

    -Routine "SYS_RESET" -Shelf "START" -Task "0"

    -Routine "RESTORE_IO" -Shelf "POWER_ON" -Task "0"
#
#
CAB_TASK_MODULES:

    -File "ram1disk:base.sys" -ModName "BASE" -Task "0"

    -File "ram1disk:user.sys" -ModName "USER" -Task "0"
#
MASTER_BOOL:

#
AUTH_INTERACTIVE_UNITS:

#
AUTH_CLIENTS:

#
AUTH_CLIENT_PERMISSION:

#
WORKDIR:

#
#
MMC_ATTRIBUTE:

#
MMC_FKEY:

#
MMC_OUTLINE:

#
```

```
#
MMC_CHOICE:

#
#
#
#
#
#
#
#
#
```

```
SYS:CFG_1.0:2:1::
#
#
SYS_MISC:

#
#
CAB_TASKS:

    -Name "MAIN" -Prog 0 -Task_in_forground "" -Type "NORMAL" -Size 0\
    -StackSize 0 -BindRef 1 -Entry "main" -RewERIfStop 0
#
CAB_REGAIN_DIST:

    -Name "MAN" -Tcp_dist 0.050000 -Tcp_rot 0.200000 -Ext_dist 0.050000\
    -Ext_rot 0.200000

    -Name "AUTO" -Tcp_dist 0.500000 -Tcp_rot 1.570000 -Ext_dist 0.500000\
    -Ext_rot 1.570000
#
CAB_EXEC_HOOKS:

    -Routine "SYS_RESET" -Shelf "RESET" -Task "0"

    -Routine "SYS_RESET" -Shelf "START" -Task "0"
```

```
    -Routine "RESTORE_IO" -Shelf "POWER_ON" -Task "0"

    -Routine "glb_power_on" -Shelf "POWER_ON" -Task "0"

    -Routine "glb_start" -Shelf "START" -Task "0"

    -Routine "glb_restart" -Shelf "RESTART" -Task "0"

    -Routine "glb_stop" -Shelf "STOP" -Task "0"

    -Routine "glb_qstop" -Shelf "QSTOP" -Task "0"

    -Routine "glb_mode_switch" -Shelf "MODE_SWITCH" -Task "0"
#
#
CAB_TASK_MODULES:

    -File "ram1disk:base.sys" -ModName "BASE" -Task "0" -Storage "LOADED"\
    -Txres 0

    -File "ram1disk:user.sys" -ModName "USER" -Task "0" -Storage "LOADED"\
    -Txres 0

    -File "ram1disk:gluser.sys" -ModName "GLUSER" -Task "0" -Storage "LOADED"\
    -Txres 0

    -File "ram1disk:glbase.sys" -ModName "GLBASE" -Task "0" -Storage "LOADED"\
    -Txres 0

    -File "ram1disk:gltext.sys" -ModName "GLTEXT" -Task "0" -Storage "LOADED"\
    -Txres 0
#
MASTER_BOOL:

#
AUTH_INTERACTIVE_UNITS:

#
AUTH_CLIENTS:
```

#
AUTH_CLIENT_PERMISSION:

#
WORKDIR:

#
#
MMC_ATTRIBUTE:

#
MMC_FKEY:

#
MMC_OUTLINE:

#
#
MMC_CHOICE:

#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#

```
#
```

```
SYS:CFG_1.0:2:0::
#
#
#
CAB_TASKS:

    -Name "MAIN" -Prog 0 -Task_in_forground "" -Type "NORMAL" -Entry "main"

    -Name "com" -Prog 1 -Task_in_forground "MAIN" -Type "SEMISTATIC"\
    -Entry "haupt"
#
CAB_REGAIN_DIST:

    -Name "MAN" -Tcp_dist 0.010000 -Tcp_rot 1.000000 -Ext_dist 0.100000\
    -Ext_rot 1.570000

    -Name "AUTO" -Tcp_dist 0.500000 -Tcp_rot 1.570000 -Ext_dist 0.500000\
    -Ext_rot 1.570000
#
CAB_EXEC_HOOKS:

    -Routine "SYS_RESET" -Shelf "RESET" -Task "0"

    -Routine "SYS_RESET" -Shelf "START" -Task "0"

    -Routine "RESTORE_IO" -Shelf "POWER_ON" -Task "0"
#
#
CAB_TASK_MODULES:

    -File "ram1disk:base_mt.sys" -ModName "BASE_MT" -Task "com"

    -File "ram1disk:user.sys" -ModName "USER" -Task "0"

    -File "ram1disk:dat_mt16.sys" -ModName "DAT_MT16" -Task "0"
```

-File "ram1disk:dat_mt16.sys" -ModName "DAT_MT16" -Task "com"

-File "ram1disk:com200.sys" -ModName "Com200" -Task "com"

-File "ram1disk:base.sys" -ModName "BASE" -Task "MAIN"

#

MASTER_BOOL:

#

AUTH_INTERACTIVE_UNITS:

#

AUTH_CLIENTS:

#

AUTH_CLIENT_PERMISSION:

#

WORKDIR:

#

#

MMC_ATTRIBUTE:

#

MMC_FKEY:

#

MMC_OUTLINE:

#

#

MMC_CHOICE:

#

#

#

#

#

#

```
#
#
#
#
#
```

SIO.CFG

Archivo:

SIO:CFG_1.0:2:0::

#

COM_PHY_CHANNEL_TYPE:

-Name "sio" -Data1 0 -Data2 0 -Data3 0 -Data4 0 -Data5 0

#

COM_PHY_CHANNEL:

-Name "sio1:" -Type "sio" -Channel 1 -Baudrate 9600 -Parity "none"\

-NoOfBits 8 -NoOfStopBits 1.000000 -Duplex "full" -Data1 0 -Data2 0\

-Data3 0 -Data4 0 -Data5 0

-Name "printer1:" -Type "sio" -Channel 2 -Baudrate 9600 -Parity "none"\

-NoOfBits 8 -NoOfStopBits 1.000000 -Duplex "full" -Data1 0 -Data2 0\

-Data3 0 -Data4 0 -Data5 0

-Name "sio3:" -Type "sio" -Channel 3 -Baudrate 9600 -Parity "none"\

-NoOfBits 8 -NoOfStopBits 1.000000 -Duplex "full" -Data1 0 -Data2 0\

-Data3 0 -Data4 0 -Data5 0

```
    -Name "sio4:" -Type "sio" -Channel 4 -Baudrate 9600 -Parity "none"\
    -NoOfBits 8 -NoOfStopBits 1.000000 -Duplex "full" -Data1 0 -Data2 0\
    -Data3 0 -Data4 0 -Data5 0
#
COM_TRP_TYPE:

    -Name "none" -Data1 0 -Data2 0 -Data3 0 -Data4 0 -Data5 0

    -Name "XON/XOFF" -Data1 0 -Data2 0 -Data3 0 -Data4 0 -Data5 0

    -Name "SLIP" -Data1 0 -Data2 0 -Data3 0 -Data4 0 -Data5 0
#
COM_TRP:

    -Name "printer1:" -Type "XON/XOFF" -PhyChannel "printer1:"\
    -HostName "none" -LocalAdress "100.100.100.101"\
    -RemoteAdress "100.100.100.102" -PortNo -1 -Data1 0 -Data2 0 -Data3 0\
    -Data4 0 -Data5 0
#
COM_APP_TYPE:

    -Name "RAP" -Data1 0 -Data2 0 -Data3 0 -Data4 0 -Data5 0 -InitFunc ""\
    -DrvC
onfig ""
```

#

COM_APP:

#

#

#

#

#

#

#

#

#

#

#

Según ChatGPT:

Este archivo es una configuración de comunicaciones serie (SIO) de un controlador ABB (RobotWare / RAPID). Define canales físicos, protocolos de transporte y aplicaciones que usan esos canales.

Voy sección por sección, de arriba a abajo.

---

1. Encabezado

SIO:CFG_1.0:2:0::

Indica:

Tipo de archivo: SIO (Serial I/O)

Versión del formato: CFG_1.0

Usado por RobotWare para configurar puertos serie RS-232/RS-422.

---

2. COM_PHY_CHANNEL_TYPE

COM_PHY_CHANNEL_TYPE:

-Name "sio"

Define un tipo de canal físico llamado "sio".

□□ Es el tipo genérico de puerto serie del controlador ABB.

---

3. COM_PHY_CHANNEL (canales físicos reales)

Aquí se definen los puertos serie disponibles y sus parámetros eléctricos.

Ejemplo:

-Name "sio1:" -Type "sio" -Channel 1 -Baudrate 9600

Parámetros clave:

Name → Nombre lógico del puerto (ej. sio1:)

Channel → Número de puerto físico (1, 2, 3, 4)

Baudrate → 9600 baudios

Parity → none

NoOfBits → 8 bits de datos

NoOfStopBits → 1 bit de parada

Duplex → full (envío y recepción)

Puertos definidos:

Puerto lógico Canal físico Uso típico

sio1: 1 Comunicación serie genérica

printer1: 2 Impresora o dispositivo externo

sio3: 3 Libre

sio4: 4 Libre

□□ Todos usan RS-232 a 9600 8N1.

---

4. COM_TRP_TYPE (tipos de protocolo)

COM_TRP_TYPE:

-Name "none"

-Name "XON/XOFF"

-Name "SLIP"

Define protocolos de transporte disponibles:

none → sin control de flujo

XON/XOFF → control de flujo por software

SLIP → encapsulación serie (menos común)

---

5. COM_TRP (transporte activo)

COM_TRP:

-Name "printer1:" -Type "XON/XOFF" -PhyChannel "printer1:"

Aquí se asigna un protocolo a un puerto físico.

Significado:

El puerto printer1:

Usa control de flujo XON/XOFF

Se apoya en el canal físico printer1:

Las direcciones IP indicadas:

-LocalAdress "100.100.100.101"

-RemoteAdress "100.100.100.102"

⚠ No son IP reales en este contexto

→ Son campos heredados del sistema de comunicaciones, no se usan en RS-232.

□□ En la práctica:

> El puerto printer1: está configurado para comunicar con una impresora o equipo serie usando XON/XOFF.

---

6. COM_APP_TYPE

COM_APP_TYPE:

-Name "RAP"

Define el tipo de aplicación:

RAP → Comunicación usada por programas RAPID

---

7. COM_APP

COM_APP:

Está vacío → no hay ninguna aplicación RAPID asociada directamente a estos puertos en este archivo.

La comunicación puede ser usada desde RAPID con instrucciones como:

Open "printer1:",\Write;

Write ...

Close;

Resumen rápido

Este archivo:

- Configura 4 puertos serie físicos
- Todos a 9600 baudios, 8N1, full duplex
- El puerto printer1: usa XON/XOFF
- Está pensado para impresora o dispositivo serie externo
- No define aún una aplicación específica, solo la infraestructura

---

Si quieres, puedo:

Decirte cómo usar printer1: desde RAPID

Explicar qué cable RS-232 usar

Ayudarte a migrarlo a RobotWare moderno

Revisar si sirve para un PLC o lector de códigos