Entering RS232 Mode

.

The RoboteQ controller can be in one of the following four Input Control Modes (ICMs):

- ICM=0 Control via a Remote Control Radio.
- ICM=1 Control via SBC connected to the RS232 port and Watchdog disabled.
- ICM=2 Control via SBC connected to the RS232 port and Watchdog enabled.
- ICM=3 Control via analog joystick.

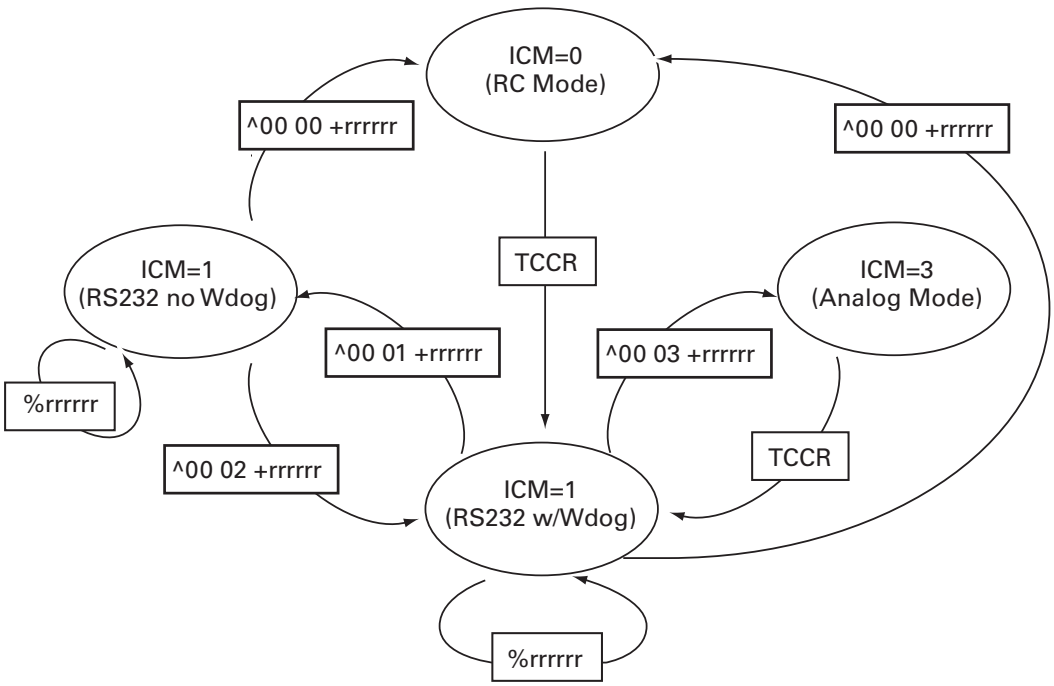
An SBC (Single Board Computer) can program a controller **only if ICM=1 or ICM=2**. If the controller is in any other ICM mode (0 or 3) it will not execute any instruction from the SBC **except** for the instruction **TCCR** (see below). Upon the SBC issuing a **TCCR** instruction the controller will switch into ICM=2.

Once the controller is in ICM=2, it can be programmed by the SBC, inclusive of the switch to any other ICM mode. Should the SBC force the controller into ICM=0 or ICM=3, the controller would no longer execute the SBC instructions, except for the TCCR which would force the controller into ICM=2, after which the controller would again execute the SBC instructions.

The diagram below illustrates how to switch from any ICM to any ICM.

TCCR is the sequence of ten consecutive carriage return characters.

%rrrrrr is the software reset, preferred to **^FF**



Physical Connection

The RoboteQ controller interfaces to an SBC via a RS232 port. The port settings must be:

9600, 7 data, 1 start, 1 stop, Even and No Flow Control

In case of an SBC running Windows, the following timings apply:

- SBC time to process an RS232 command: **1 millisecond**. This is as fast as the SBC can push commands through the RS232 no matter what is connected to it.
- RoboteQ controller update frequency: **every 16 millisecond**. The controller applies a new motor command every 16 ms. The SBC should not submit a new position instructions to the RoboteQ controller faster than 16 ms, so it is advisable to append a 20 ms delay after any instruction or use a timer with a 20 ms time base.
- Should the SBC be programmed in a such a way that it has to submit instructions in packets, then an instruction buffer should take care of adapting the timing of the SBC program with the timing of the RoboteQ controller.

Instruction Set

All instructions to RoboteQ controller are character strings sent via the RS232 port, and terminated by a carriage return.

The controller will echo back special codes to signify acceptance or rejection of the instruction (rejection for bad instruction or bad data in instruction). If the controller rejects an instruction, such instruction must be considered lost and therefore it needs to be corrected and repeated.

Instructions are divided in few basic categories:

1. PRESET MODIFY instruction

A Preset Modify is an instruction which assigns to the preset parameter WW one of its possible values.

Structure: **^WW XX** where **WW** is the parameter and **XX** its value.
The space between **WW** and **XX** is compulsory.

A PRESET is received correctly if the controller echoes back: **^WW XX+**

Echoing back **^WW XX-** indicates that either **WW** or **XX** are outside the permitted range of values.

Most Preset instructions take effect after the special instruction Reset, %rrrrrr.

Here is the sequence to make a change to the value of a parameter:

SBC:	**^WW XX**	change parameter **WW** to new value **XX**
Controller:	**^WW XX+**	+ means acceptance of previous instruction
SBC:	**%rrrrrr**	make change effective
Controller:	Reset string	change effected

2. PRESET READ instruction

A Preset Read is an instruction that reads the value XX of the preset parameter WW. The parameter WW is not modified; only its value XX is echoed back.

Structure: **^WW**

The controller echoes back **^WW XX+**

3. COMMAND instruction

A Command is an instruction forcing the controller to apply a value YY to an output parameter O.

Structure: **!OYY** where **O** is the output and **YY** its hex value.

The hex value YY can be signed or unsigned according to the output in question.

In case the output O is boolean (example controlling a digital output) the Command structure simply becomes:

Structure: **!O or !o** where **O**=ON and **o**=off

A Command is received correctly if the controller echoes back the Command followed by **+**, (example **!OYY+**). Should the **O** or **YY** be outside of the range of permitted values, the controller will echo back the Command followed by **-** where the minus sign signifies that the controller has not executed the Command (example: **!OYY-**).

4. QUERY instruction

A Query is an instruction requesting the controller to echo back the value of a sensor or an input/output parameter.

Structure: **?W** where **W** is the parameter being queried.

The RoboteQ controller will echo back: **?WXXYY** if two values will be returned. **?WXXYYZZ** if three values will be returned. **XX**, **YY**, **ZZ** are hex values ranging from 00h to FFh.

A Query is received correctly if the controller echoes back **?WXXYY** or **?WXXYYZZ**. Note the absence of the + confirming receipt.

In case of an out of range query, the reply will be **?W-**

5. OTHER instructions

RESET %rrrrrr

Reset is received correctly if the controller echoes back the reset string.

PRESET EFFECTIVE ^FF

Preset Effective is received correctly if the controller echoes back ^FF+

(**TCCR**) 10 consecutive carriage returns chr(13) x ten times

If **ICM = 0** or **ICM =3** then **TCCR** forces the controller into **ICM=2**.

If **ICM=1** or I**CM=2** then **TCCR** does not change the existing value of **ICM**.

Examples

Assuming Visual Basic 6.0 under Windows inclusive of an RS232 control with functions SEND and RECEIVE, typical instruction sequences would be:

```
'Main code
Dim voltages as string
  'Comment: Adjust power to motor1 to the value 10
   Call Power("10")
  'Comment: Change ICM to value 2
   Call Change_ICM("02")
   'Comment: Query Voltages
   voltages = Query_voltages
   'End Main

Private sub Power(value as string)
Dim strng as string
   Strng = "!A " + value + VBCR     'Comment: keep space after !A
   SEND = strng
   End sub

Private sub Change_ICM(mode as string)
Dim strng as string
   Strng = "^00 " + mode + VBCR    'Comment:keep space after ^00
   SEND strng
   End sub

Private function Query_voltages as string
Dim strng as string
   Strng = "?E"
   SEND = strng
   QUERY_voltages = RECEIVE
   End sub
```