MotionChip™ II

TECHNOSOFT

Data Sheet

Preliminary

MotionChip™ II
Data Sheet

P091.055.MCII.DSH.1104
November 2004

Technosoft S.A.
Rue des Courtils 8A
CH 2035 Corcelles-NE
Switzerland
Tel.: +41 (0) 32 732 5500
Fax: +41 (0) 32 732 5504
contact@technosoftmotion.com
www.technosoftmotion.com/

If you want to …	Contact Technosoft at …
Visit Technosoft online	World Wide Web: http://www.technosoftmotion.com/
Receive general information or assistance Ask questions about product operation or report suspected problems Make suggestions about or report errors in documentation	World Wide Web: http://www.technosoftmotion.com/ Email: contact@technosoftmotion.com Fax: (41) 32 732 55 04 Email: hotline@technosoftmotion.com Mail: Technosoft SA Case postale 250 Rue des Courtils 8a CH 2035 Corcelles-NE Switzerland

Contents

Figures

Tables

1. Features

- High-performance ready to run DSP Motion Controller, 40MHz, 40MIPS
- Single-Chip solution for control of
  - DC Brush Motors
  - DC Brushless Motors
  - AC Brushless Motors (PMSM)
- Operation Modes
  - Stand-alone – executes motion sequences from a local memory
  - Slave – multi-axis cases
- Communication channels:
  - Serial RS-232 / RS-485
  - CAN-bus
- Flexible structure allowing:
  - Position control
  - Speed control
  - Torque control (field-oriented for AC motors)
  - Open-loop (V/f for AC motors)
- Typical sampling rates: 10kHz torque loop, 1kHz speed/position loop
- Feedback signals
  - 1- 3 Currents
  - Hall sensors
  - Position read from incremental encoder (on-chip interface)
  - Speed estimated from position
  - Two temperature sensors
  - DC-bus voltage (V_{DC})
- High-level programming using the Technosoft Motion Language (TML)
- No DSP Code Development Required
- 23 motion modes, decision blocks, functions, arithmetic & logic unit
- Accurate profile generator with automatic round-off correction
  - Position range: 32-bits
  - Speed/acceleration range: 16-bit integer part, 16-bit fractional part
- 18 programmable events were motion parameters and/or modes can be changed on-the-fly
- 12 programmable TML interrupts
- Main Inputs:
  - Power Drive Fault (interrupt input)
  - Enable/Disable (interrupt input)
  - 2 Limit switches (interrupt inputs)
  - Encoder Index (capture input)
  - 2nd Encoder Index (capture input)
  - Start mode: AUTORUN / Wait commands
- Main Outputs:
  - 4-6 PWM commands
  - Brake transistor command
  - Interrupt to host
  - Ready
- Up to 19 general-purpose I/O
- Advanced PWM command methods
  - V_{DC} variation compensation;
  - Dead-time compensation;
  - 3rd harmonic injection;
  - Wobbling for EMI reduction
- Brake transistor control
- Integrated protections: over current, over voltage, under voltage, over temperature, I^2t,
- Development tools available:
  - IPM Motion Studio including:
    - Motor and drive configuration dialogs;
    - Motion sequence setup wizard;
    - Compiler, command interpreter;
    - Graphical display for traced variables, watch functions.
  - Application notes (web page)
  - Reference designs (web page)

2. General Description

The Technosoft **MotionChip™ II** is a high performance ready-to-run motion controller, based on the latest DSP structures. It does not require any DSP Code development. With the advantage to contain all necessary configurable interfaces on a single chip, the MotionChip II is ideal for rapid and cost effective design of fully digital, intelligent drives for various motor types.

The MotionChip II can:

- Operate in a stand-alone or in a master / slave, multiple axis configuration
- Control three motor types: DC brush, DC brushless, AC brushless (PMSM)
- Implement various command structures: open loop, torque, speed, position / external loop control
- Work with different motion and protection sensors (position, speed, current, torque, voltage, temperature)
- Use different communication channels such as SCI for RS232/485 and CAN-bus
- Execute advanced motion language commands and motion sequences

Intended to cover a major part of basic and complex motion applications, the MotionChip II has the special advantage to be a highly flexible structure at the level of:

- Motion structure configuration (selection of motor technology, control type, sensors type)
- Motion implementation with high-level motion language commands

Relative to existing solutions the MotionChip II offers many advantages:

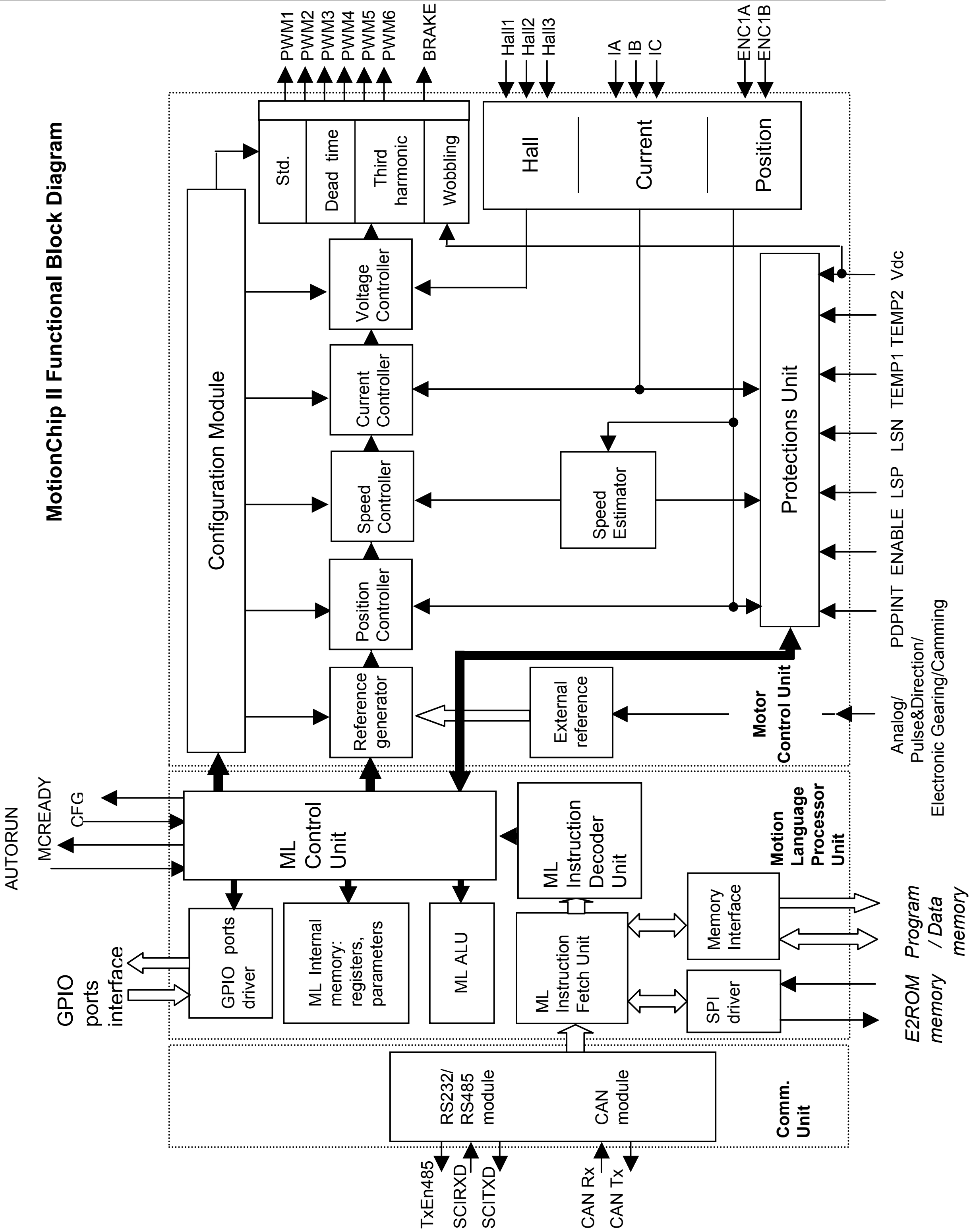
- Usable for different motor technologies
- Implementation of multiple motion control configurations, including vector control for AC drives
- Implementation of complete digital control loops, including current/torque control
- Powerful motion language including 23 motion modes, decision blocks, function calls, event-driven motion updates, interrupts
- Stand-alone or slave operation
- Minimal requirements for setup configuration and use
- Easy embedding in user hardware structures
- Software-less device (no programming effort required)
- High-level development tools for application setup, test and debug

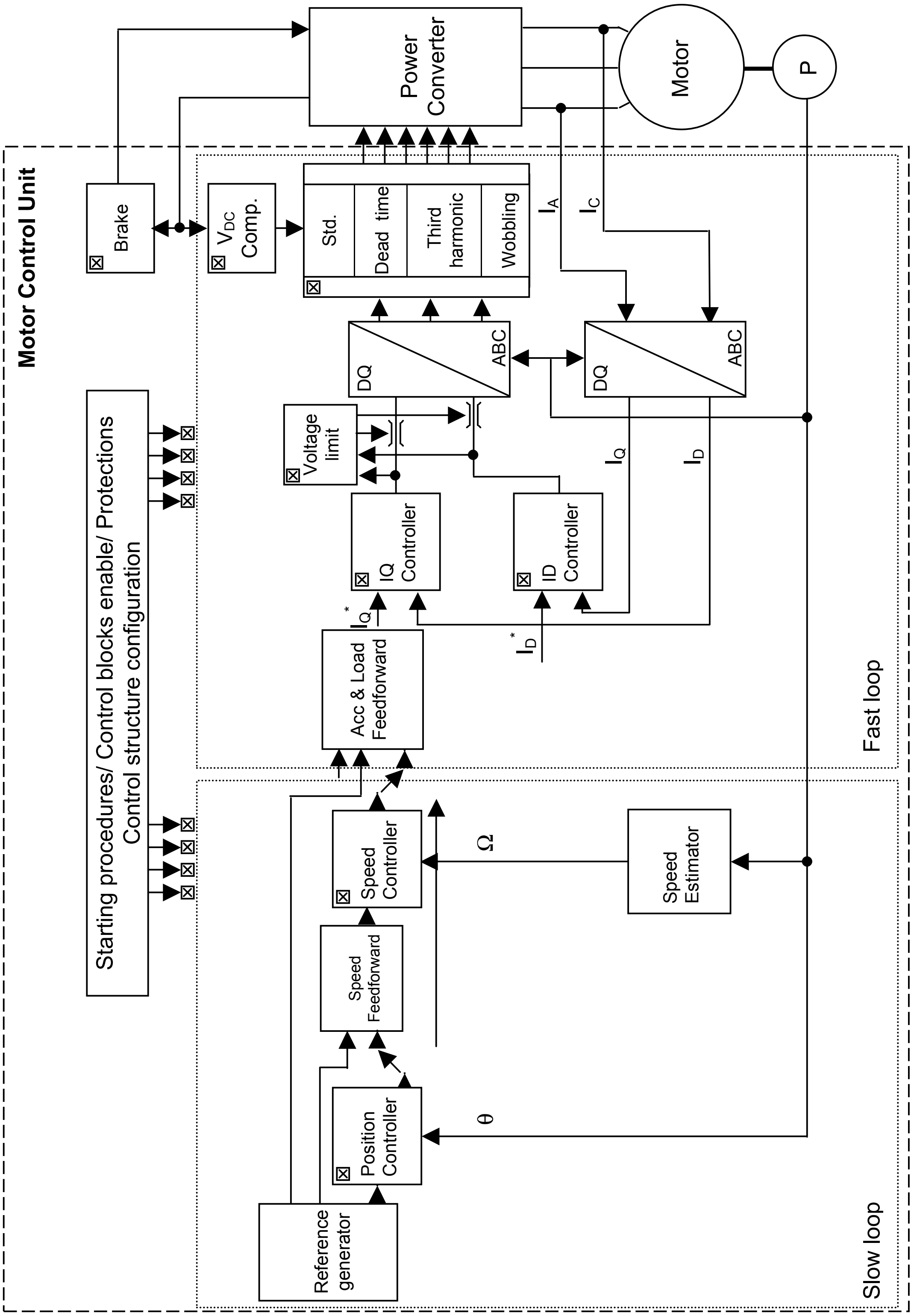
2.1. Functional Block Diagram

The functional block diagram provides a high-level description of the basic components of the device. The MotionChip II has three functional units:

- the communication unit, through which the MotionChip II can receive and send motion language commands using serial (RS-232 or RS-485) or CAN-bus;
- the "motion language processor" unit which decodes motion language commands received from communication unit or fetches them from a local memory
- the motor control unit that executes the motion commands. The motor control unit diagram presents in detail the components of this unit.

MotionChip II Functional Block Diagram





Motor Control Unit
Power Converter
Motor
P
Brake
V_{DC} Comp.
Std.
Dead time
Third harmonic
Wobbling
I_A
I_C
DQ
ABC
DQ
ABC
Voltage limit
IQ Controller
ID Controller
I_Q
I_D
I_Q^*
I_D^*
Acc & Load Feedforward
Starting procedures/ Control blocks enable/ Protections
Control structure configuration
Fast loop
Speed Controller
Ω
Speed Estimator
Speed Feedforward
Position Controller
θ
Reference generator
Slow loop

2.2. Motion Configurations

MotionChip II is designed to control brushless motors and brush DC motors.
Brushless motors are vector controlled and can be driven with sinusoidal commutation (PMSM) or with trapezoidal commutation (BLDC).

Each type of motor can be controlled in position, speed and current/torque closed loop or can be commanded in open loop.

The MotionChip II provides 23 motion modes. Each motion mode designates a reference mode and a control structure. Table 2.1 summarizes for each type of control the reference modes accepted.

Table 2.1. *Motion modes*

Motion Modes	**Control Type**			
	Position	**Speed**	**Torque**	**Voltage**
Position Profiles	√	–	–	–
Speed Profiles	–	√	–	–
Contouring (point to point with linear interpolation)	√	√	√	√
External, reference read from the analogue input REFERENCE	√	√	√ SL √FL	√ SL √ SL
Pulse and direction	√	√	–	–
Electronic Gearing/Camming – master	√	–	–	–
Electronic Gearing – slave	√	√	–	–
Electronic Camming – slave	√	–	–	–
Stop	–	√	√	√
Test (limited ramp)	–	–	√FL	√FL

Legend:
SL – Reference update is done on the slow (position/speed) loop
FL – Reference update is done on the fast (current/torque) loop

2.3. Controlled Loops

The MotionChip II control structure includes three control loops. The outer loop is used for motor position control. The outer loop controller is a PID with filter on the derivative term. The PID output can be used as speed or torque command for the motor.

The middle loop implements the speed control. The speed loop controller is a PI with speed feedforward. The speed loop controller output it is used as torque command for the motor.

The inner loop performs the current/torque control. For PMSM motors, torque control is performed using a field oriented control scheme. The inner loop has two PI controllers one for torque control (Q axis controller) and the other for flux control (D axis controller). The inner loop provides a voltage command, which is translated into PWM commands.

2.4. Sensors

Table 2.2 presents the categories of sensors accepted by the MotionChip II, where are they used, their type and the required interface or connections.

Table 2.2. *MotionChip II accepted sensors*

Category	Usage	Type	Interface/Connections
Position	Position feedback	Incremental encoder	On-chip. Dedicated inputs: ENC1A for A, ENC1B for B, ENC1Z for Index (if available)
Speed	Speed feedback	Computed from position	The position sensor interface
Current	Current feedback	1/2 current transducers on motor lines or lower legs of the inverter	Dedicated inputs: IA, IC
Temperature	Motor and/or power stage protection	1/2 analogue sensors	Dedicated inputs: TEMP1 and TEMP2
Hall	Commutation	120° apart	Dedicated inputs: HALL1, HALL2, HALL3
Voltage	Over-voltage and under-voltage protections. PWM control with VDC compensation	Analogue sensor	Dedicated input: VDC

2.5. Special Features

The MotionChip II includes a set of programmable special features. These offer the possibility to select different strategies according with the application specific. Table 2.3 summarizes these features.

Table 2.3. *Special features*

Function	Options
Current Offset Detection	Automatic, with motor supplied (PWM outputs active)
	Automatic, without motor supplied (PWM outputs inactive)
	Automatic detection is disabled. User provides the offset
Start method for the brushless AC (PMSM) motors	Motor is aligned on phase A, by injecting a current in phases B and A
	Motor is aligned on phase A, by applying a voltage on phases B and A
	Motor starts as a brushless DC using Hall commutation. After first Hall transition, commutes to brushless AC mode (FOC, sinusoidal currents)
PWM command Techniques	With compensation of DC-bus voltage variation (needs DC-bus voltage sensor)
	With compensation of dead-time
	With 3rd harmonic injection to increase maximum applicable voltage
	With PWM frequency wobbling around programmed value to reduce EMI (electromagnetic interference)

Brake command	DC-bus voltage control during motor brakes, by command of a brake transistor

PWM Frequency and Sampling Rates

The MotionChip II uses a fast loop for current/torque control and a slow loop for position/speed sampling. The sampling rates of these loops are synchronized and linked in a fixed ratio with the PWM frequency in order to eliminate the beat-frequency problems. The maximum sampling frequency on the fast loop can be half of the PWM frequency. The PWM frequency and the divider ratios for fast and slow loops are user programmable in a wide range. The maximum values for PWM frequency and sampling rate frequencies depend on the application configuration. Table 2.4 shows the typical (default) values, which cover all motion application configurations.

Table 2.4.*Typical values for PWM frequency and sampling rates*

PWM	Fast loop (current/torque)	Slow loop (position/speed)
20kHz	10kHz	1kHz

3. Electrical and mechanical specifications

The MotionChip II is designed to be programmed on two versions of DSP made by Texas Instruments: TMS320LF2407A or TMS320LF2406A.

The next sections present MotionChip II specific connections with supported peripherals and memory.

All electrical and mechanical specifications can be found in TMS320LF2407A/TMS320LF2406A Data Sheet (literature number SPRS145i) downloadable from Texas Instruments web page www.ti.com.

Table 3.1 describes the correspondence between DSP pins and MotionChip II pins.

Table 3.1. *Pins correspondence*

DSP pin name	Pin Number		MotionChip™ II name	Description
	LF2407A	LF2406A		
Event manager A (EVA)				
CAP1/QEP1/***IOPA3***	83	57	IO # 3/ENC1A	A signal from encoder
CAP2/QEP2/***IOPA4***	79	55	IO # 4/ENC1B	B signal from encoder
CAP3/***IOPA5***	75	52	IO # 5/CAPI	Z signal from encoder or GPIO
PWM1/***IOPA6***	56	39	PWM1	PWM1 command
PWM2/***IOPA7***	54	37	PWM2	PWM2 command
PWM3/***IOPB0***	52	36	PWM3	PWM3 command
PWM4/***IOPB1***	47	33	PWM4	PWM4 command
PWM5/***IOPB2***	44	31	PWM5	PWM5 command
PWM6/***IOPB3***	40	28	PWM6	PWM6 command
T1PWM/T1CMP/***IOPB4***	16	12	ERROR	ERROR signal
T2PWM/T2CMP/***IOPB5***	18	13	IO # 13	GPIO
TDIRA/***IOPB6***	14	11	IO # 14	GPIO
TCLKINA/***IOPB7***	37	26	IO # 15	GPIO
Event manager B (EVB)				
CAP4/QEP3/***IOPE7***	88	60	IO # 32	Third Hall sensor input or GPIO
CAP5/QEP4/***IOPF0***	81	56	IO # 33	First Hall sensor input or GPIO
CAP6/***IOPF1***	69	48	IO # 34	Second Hall sensor input or GPIO
PWM7/***IOPE1***	65	45	IO # 26 /BRAKE	BRAKE command output or GPIO
PWM8/***IOPE2***	62	43	IO # 27	GPIO
PWM9/***IOPE3***	59	41	IO # 28	GPIO
PWM10/***IOPE4***	55	38	IO # 29	GPIO
PWM11/***IOPE5***	46	32	IO # 30	GPIO
PWM12/***IOPE6***	38	27	IO # 31	GPIO
T3PWM/T3CMP/***IOPF2***	8	7	IO # 35	Third Hall sensor input or GPIO
T4PWM/T4CMP/***IOPF3***	6	5	IO # 36	GPIO
TDIRB/***IOPF4***	2	2	IO # 37/DIR	Direction signal for P&D mode or GPIO
TCLKINB/***IOPF5***	126	89	IO # 38/PULSE	Pulse signal for P&D mode or GPIO

ANALOG-TO-DIGITAL CONVERTER (ADC)				
ADCIN00	112	79	AD6	Not used, connect to the ground
ADCIN01	110	77	AD7	Drive temperature
ADCIN02	107	74	AD2	Not used, connect to the ground
ADCIN03	105	72	AD5	Analogue reference
ADCIN04	103	70	AD4	Motor supply voltage
ADCIN05	102	69	AD1	Current in phase C
ADCIN06	100	67	AD0	Current in phase A
ADCIN07	99	66	AD3	Not used, connect to the ground
ADCIN08	113	80	–	Select RS485 (0)/RS232 (1)
ADCIN09	111	78	–	Select AUTORUN (0)/slave mode (1)
ADCIN10	109	76	–	Axis ID 5th bit
ADCIN11	108	75	–	Axis ID 4th bit
ADCIN12	106	73	–	Axis ID 3rd bit
ADCIN13	104	71	–	Axis ID 2nd bit
ADCIN14	101	68	–	Axis ID 1st bit
ADCIN15	98	65	–	Motor temperature
V_{REFHI}	115	82	–	ADC analog high-voltage reference input
V_{REFLO}	114	81	–	ADC analog low-voltage reference input
V_{CCA}	116	83	–	Analog supply voltage for ADC (3.3 V)
V_{SSA}	117	84	–	Analog ground reference for ADC
CONTROLLER AREA NETWORK (CAN), SERIAL COMMUNICATIONS INTERFACE (SCI), SERIAL PERIPHERAL INTERFACE (SPI)				
CANRX/***IOPC7***	70	49	IO # 23	CAN receive data or GPIO
CANTX/***IOPC6***	72	50	IO # 22	CAN transmit data or GPIO
SCITXD/***IOPA0***	25	17	SCITxD	SCI asynchronous serial port transmit data
SCIRX/***IOPA1***	26	18	SCIRxD	SCI asynchronous serial port receive data
SPICLK/***IOPC4***	35	24	IO # 20	SPI clock
SPISIMO/***IOPC2***	30	21	IO # 18	SPI slave in, master out
SPISOMI/***IOPC3***	32	22	IO # 19	SPI slave out, master in
$\overline{\text{SPISTE}}$ /***IOPC5***	33	23	IO # 21	SPI slave transmit-enable
EXTERNAL INTERRUPTS, CLOCK				
$\overline{\text{RS}}$	133	93	–	Device Reset (in) and Watchdog Reset (out).
$\overline{\text{PDPINTA}}$	7	6	–	Power drive protection interrupt input.
XINT1/***IOPA2***	23	16	IO # 2/LSP	Limit Switch Positive or GPIO
XINT2/ADCSOC/***IOPD0***	21	15	IO # 24/LSN	Limit Switch Negative or GPIO
CLKOUT/IOPE0	73	51	READY	Drive READY signal
$\overline{\text{PDPINTB}}$	137	95	–	Power drive protection interrupt input. This interrupt, when activated, puts the PWM output pins (EVB) in the high-impedance state
OSCILLATOR, PLL, FLASH, BOOT, AND MISCELLANEOUS				
XTAL1/CLKIN	123	87	–	PLL oscillator input pin. Crystal input to PLL/clock source input to PLL. XTAL1/CLKIN is tied to one side of a reference crystal.
XTAL2	124	88	–	Crystal output. PLL oscillator output pin.

PLLVCCA	12	10	–	PLL supply (3.3 V)
IOPF6	131	92	–	General-purpose I/O
BOOT_EN /XF	121	86	–	Boot ROM enable, GPO, XF. This pin will be sampled as input (BOOT_EN) to update SCSR2.3 (BOOT_EN bit) during reset and then driven as an output signal for XF.
PLLF	11	9	–	PLL loop filter input 1
PLLF2	10	8	–	PLL loop filter input 2
VCCP (5V)	58	40	–	Flash programming voltage pin. This pin must be connected to a 5-V supply for Flash programming.
TP1	60	42	–	Test pin 1. *Do not connect.*
TP2	63	44	–	Test pin 2. *Do not connect.*
BIO /IOPC1	119	85	IO # 17/485TxEn	485TxEn: "0"=receive, "1"=transmit or GPIO
EMULATION AND TEST				
EMU0	90	61		Emulator I/O #0 with internal pullup.
EMU1/OFF	91	62	–	Emulator pin 1. Emulator pin 1 disables all outputs. EMU1/OFF, when active low, puts all output drivers in the high-impedance state.
TCK	135	94	–	JTAG test clock with internal pullup
TDI	139	96	–	JTAG test data input (TDI) with internal pullup.
TDO	142	99	–	JTAG scan out, test data output (TDO).
TMS	144	100	–	JTAG test-mode select (TMS) with internal pullup.
TMS2	36	25	–	JTAG test-mode select 2 (TMS2) with internal pullup.
TRST	1	1	–	JTAG test reset with internal pulldown. TRST, when driven high, gives the scan system control of the operations of the device. If this signal is not connected or driven low, the device operates in its functional mode, and the test reset signals are ignored.
ADDRESS, DATA, AND MEMORY CONTROL SIGNALS				
DS	87	–	–	Data space strobe. IS, DS, and PS are always high unless low-level asserted for access to the relevant external memory space or I/O. They are placed in the high-impedance state.
IS	82	–	–	I/O space strobe. IS, DS, and PS are always high unless low-level asserted for access to the relevant external memory space or I/O. They are placed in the high-impedance state.
PS	84	–	–	Program space strobe. IS, DS, and PS are always high unless low-level asserted for access to the relevant external memory space or I/O. They are placed in the high-impedance state.
R/ W	92	–	–	Read/write qualifier signal. R/W indicates transfer direction during communication to an external device. It is normally in read mode (high), unless low level is asserted for performing a write operation. R/W is placed in the high-impedance state.
W / R / IOPC0	19	14	ENABLE	ENABLE input signal
RD	93	–	–	Read-enable strobe. Read-select indicates an active, external read cycle. RD is active on all external program, data, and I /O reads. RD is placed in the high-impedance state.

$\overline{WE}$	89	–	–	Write-enable strobe. The falling edge of WE indicates that the device is driving the external data bus (D15-D0). WE is active on all external program, data, and I/O writes. WE is placed in the high-impedance state.
$\overline{STRB}$	96	–	–	External memory access strobe. STRB is always high unless asserted low to indicate an external bus cycle. STRB is active for all off-chip accesses. STRB is placed in the high-impedance state.
READY	120	–	–	READY is pulled low to add wait states for external accesses. READY indicates that an external device is prepared for a bus transaction to be completed. If the device is not ready, it pulls the READY pin low. The processor waits one cycle and checks READY again.
MP/ $\overline{MC}$	118	–	–	Microprocessor/Microcomputer mode select.
ENA_144	122	–	–	Active high to enable external interface signals.
$\overline{VIS\_OE}$	97	–	–	Visibility output enable (active when data bus is output).
A0	80	–	–	Bit 0 of the 16-bit address bus
A1	78	–	–	Bit 1 of the 16-bit address bus
A2	74	–	–	Bit 2 of the 16-bit address bus
A3	71	–	–	Bit 3 of the 16-bit address bus
A4	68	–	–	Bit 4 of the 16-bit address bus
A5	64	–	–	Bit 5 of the 16-bit address bus
A6	61	–	–	Bit 6 of the 16-bit address bus
A7	57	–	–	Bit 7 of the 16-bit address bus
A8	53	–	–	Bit 8 of the 16-bit address bus
A9	51	–	–	Bit 9 of the 16-bit address bus
A10	48	–	–	Bit 10 of the 16-bit address bus
A11	45	–	–	Bit 11 of the 16-bit address bus
A12	43	–	–	Bit 12 of the 16-bit address bus
A13	39	–	–	Bit 13 of the 16-bit address bus
A14	34	–	–	Bit 14 of the 16-bit address bus
A15	31	–	–	Bit 15 of the 16-bit address bus
D0	127	–	–	Bit 0 of the 16-bit data bus
D1	130	–	–	Bit 1 of the 16-bit data bus
D2	132	–	–	Bit 2 of the 16-bit data bus
D3	134	–	–	Bit 3 of the 16-bit data bus
D4	136	–	–	Bit 4 of the 16-bit data bus
D5	138	–	–	Bit 5 of the 16-bit data bus
D6	143	–	–	Bit 6 of the 16-bit data bus
D7	5	–	–	Bit 7 of the 16-bit data bus
D8	9	–	–	Bit 8 of the 16-bit data bus
D9	13	–	–	Bit 9 of the 16-bit data bus
D10	15	–	–	Bit 10 of the 16-bit data bus
D11	17	–	–	Bit 11 of the 16-bit data bus
D12	20	–	–	Bit 12 of the 16-bit data bus

D13	22	–	–	Bit 13 of the 16-bit data bus
D14	24	–	–	Bit 14 of the 16-bit data bus
D15	27	–	–	Bit 15 of the 16-bit data bus
POWER SUPPLY				
VDD	29	20	–	Core supply +3.3 V. Digital logic supply voltage.
	50	35	–	
	86	59	–	
	129	91	–	
VDDO	4	4	–	I/O buffer supply +3.3 V. Digital logic and buffer supply voltage.
	42	30	–	
	67	47	–	
	77	54	–	
	95	64	–	
	141	98	–	
VSS	28	19	–	Core ground. Digital logic ground reference.
	49	34	–	
	85	58	–	
	128	90	–	
VSSO	3	3	–	I/O buffer ground. Digital logic and buffer ground reference.
	41	29	–	
	66	46	–	
	76	53	–	
	94	63	–	
	125	97	–	
	140	–	–	

4. MotionChip II I/O

4.1. Predefined Digital Outputs

4.1.1. PWM Outputs

The MotionChip II provides 6 PWM outputs, 3 active low (PWM2, PWM4 and PWM6) and 3 active high (PWM1, PWM2 and PWM3).

In application configurations with three-phase motors (DC brushless and PMSM), each of the six PWM outputs drives one switching device of a three-phase inverter. PWM1 (PWM1/IOPA6), PWM3 (PWM3/IOPB0) and PWM5 (PWM5/IOPB2) drive the upper switching devices. PWM2 (PWM2/IOPA7), PWM4 (PWM4/IOPB1) and PWM6 (PWM6/IOPB3) drive the lower switching devices, see Figure 4.1.

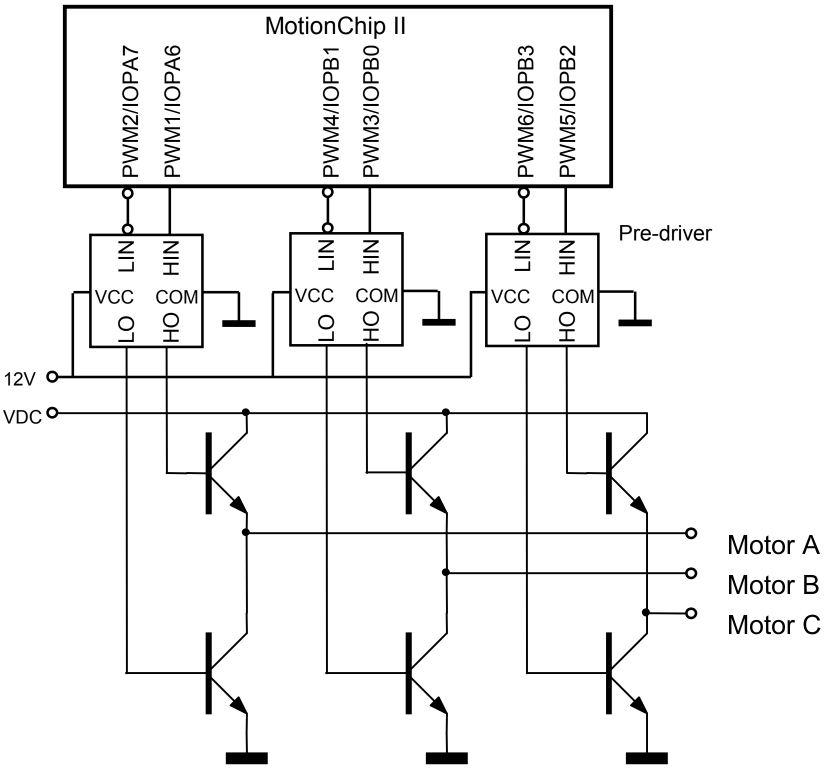


Figure 4.1. *PWM outputs with three phase inverter*

In application configurations with DC motors, only 4 PWM outputs are used, each drives one switching device of an H-bridge inverter. PWM1/IOPA6 and PWM3/IOPB2 drive the upper switching devices. PWM2/IOPA7 and PWM4/IOPB1 drive the lower switching devices. PWM1/IOPA6& PWM2/IOPA7 drive one bridge leg, PWM3/IOPB2& PWM4/IOPB1 drive the other leg (see Figure 4.2). PWM5/IOPB2 & PWM6/IOPB3 are not used and are set high (inactive); they may be left unconnected for application configurations with DC motors.

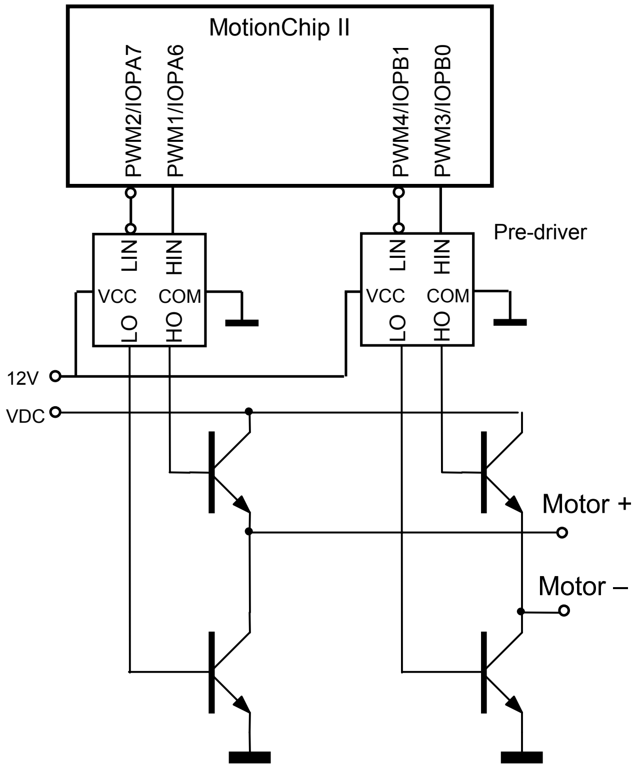


Figure 4.2. *PWM outputs usage with H bridges (two-phase inverters)*

At power-on or after any MotionChip II reset, all PWM outputs remain in the high-Z state until TML command `AXISON` is first executed. PWM outputs go immediately to the high-impedance state in one of the following cases:

- when $\overline{\text{PDPINT}}$ input goes low (high to low transition)
- when ENABLE input goes low (high to low transition) and as long as ENABLE remains low
- when TML command `AXISOFF` is executed;

The MotionChip II provides a programmable dead-band for each PWM output pair (1-2, 3-4, 5-6) from 0 to 12μs. The default value is 0μs. Hence you must set the TML parameter DBT with a nonzero value or use external dead-band logic in order to drive the power stage of a three-phase motor or that of a DC brush motor i.e. the configurations where each PWM output drives one switching device. For more details about power converter setup see MotionChip II Configuration Setup User Manual.

4.1.2. BRAKE Output

The MotionChip II can monitor the DC-bus voltage level if a voltage feedback is provided on the analogue input VDC. In some cases, during motor brakes, the DC-bus voltage may grow to dangerous values due to the current injected by the motor into the DC-link. One solution to reduce this over-voltage is to connect temporary a resistor across the DC-link through a switching device (see Figure 4.3). The MotionChip II has an active-high PWM output called BRAKE (PWM7/IOPE1) dedicated to drive the switching device from a brake circuit. When this function is enabled, the BRAKE output is automatically activated with a 50% duty cycle, if the DC-bus voltage bypasses a programmed trigger value. The BRAKE output remains active until the DC-bus voltage drops under this trigger value. The activation of the BRAKE command can be signaled by a TML interrupt (see MotionChip™ II TML Programming User Manual) if the over-voltage protection limit is set at the same value as the BRAKE trigger level.

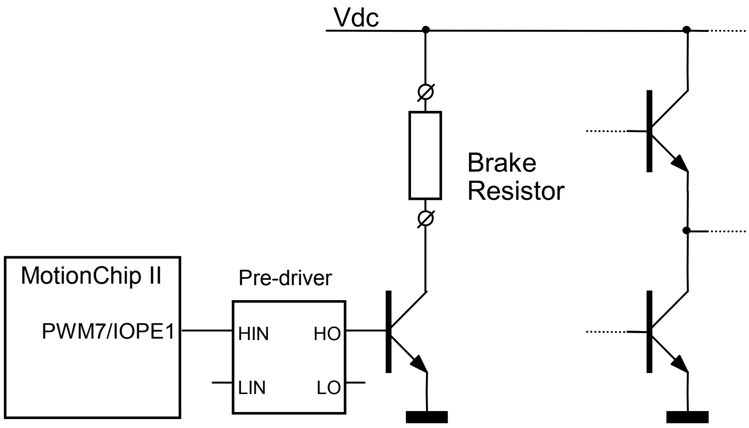


Figure 4.3. *BRAKE pin usage*

4.1.3. READY Output

The READY (CLKOUT/IOPE0) output is set high immediately after reset. It goes low after the MotionChip II internal initialization ends. This output can be used to signal to an external device that the MotionChip is powered and is ready to receive and execute TML commands.

4.2. Predefined Digital Inputs

4.2.1. $\overline{\text{PDPINT}}$ – Power Drive Protection Interrupt Input

$\overline{\text{PDPINT}}$ ($\overline{\text{PDPINTA}}$) is an interrupt input that immediately shuts-down (deactivates) the PWM outputs when high to low transition occurs on this pin. The goal of this input is to offer quick reacting protection to the power stage if at this level an error occurs. When the falling edge is sensed, all 6 PWM outputs plus the BRAKE signal (if activated) are set to High-Z state.

4.2.2. LSP, LSN - Limit Switch Inputs

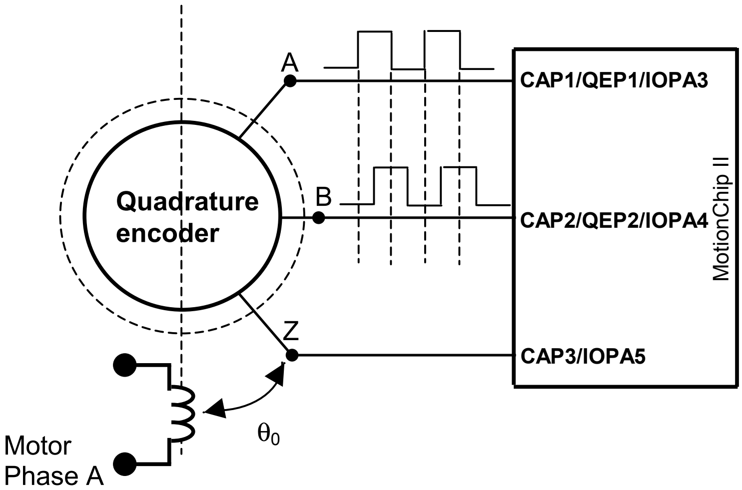
The MotionChip II has two dedicated interrupt inputs for limit switches: Limit Switch Positive (XINT1/IOPA) and Limit Switch Negative (XINT2/IOPD). The inputs can be programmed to detect transitions from low to high or high to low. The following actions can be programmed when a transition is detected on a limit switch input:

- Generate a TML interrupt. The TML interrupt service routine accepts any TML command, hence when a limit switch is reached, any desired action can be programmed;
- Generate a TML event. On event occurrence, the motion mode and/or parameters can be automatically changed. This feature is useful if the motor is on-purpose moved towards a limit switch, with a goal to execute a specific motion sequence when the limit switch is reached.

4.2.3. Quadrature Encoder Inputs

The MotionChip II includes two incremental encoder interfaces. The first interface is used to read motor position (CAP1/QEP1/IOPA3, CAP2/QEP2/IOPA4, CAP3/IOPA5), while the second interface (CAP4/QEP3/IOPE7, CAP5/QEP4/IOPF0, CAP6/IOPF1) is used to read master position from master's second encoder.

If the encoder quadrature signals A and B are CMOS level compatible, they can be connected directly to the MotionChip II inputs ENC1A and ENC1B; else the signals level must be adapted. The position is incremented or decremented by the rising or falling edge of the two input signals (four times the frequency of either input pulse). The minimum pulse length on any encoder input is 112ns.



Remark *The second encoder inputs are used also for Hall sensors inputs, so the use of one type of sensor excludes the other one.*

The second encoder can't be used with DC brushless motors, since they require feedback from Hall sensors.

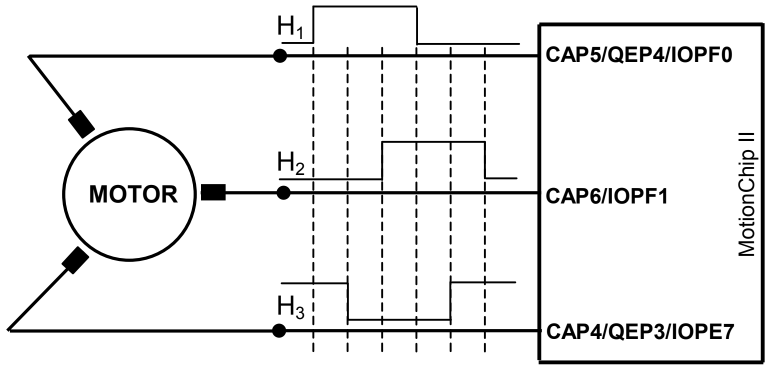
4.2.4. CAPI, CAPI2 - Capture Inputs

Digital inputs IO#5/CAPI (CAP3/IOPA5) and IO#34/2CAPI (CAP6/IOPF1) can be used as capture inputs they can be programmed to sense either a low to high or high to low transition. When the programmed transition occurs, the motor position or the master position is captured e.g. memorized in dedicated variables. These are CAPPOS for the motor position and CAPPOS2 for the master position. The captured position is very accurate as the whole process is very fast (less than 200 ns if the position sensor is an encoder).

4.2.5. HALL Sensors Inputs

The Hall sensors signals coming from a brushless DC motor can be directly connected to MotionChip II HALL1 (CAP4/QEP3/IOPE7), HALL2 (CAP6/IOPF1) and HALL3 (CAP5/QEP4/IOPF0) inputs if the signals are CMOS level compatible; else the signals level must be adapted.

The MotionChip II can work with Hall signals phased at 120 electrical degrees. In the relation between the voltage succession and Hall sensors values, there are 12 possible cases. The MotionChip II supports all, hence there is no restriction concerning the Hall sensors sequence or polarity. In application that doesn't require Hall sensors feedback this inputs can be used to read master position from the second quadrature encoder.



Remark: *In a motion application based on a brushless DC motor, these pins will be automatically set as inputs to read Hall signals, when `ENDINIT` TML instruction is executed.*

The second encoder can not be used in motion application with brushless DC motor because uses the same pins as Hall sensors.

4.2.6. ENABLE Input

ENABLE ($W/\overline{R}$ /IOPC0) input is an external emergency stop. When asserted high, the MotionChip II executes the TML command `AXISOFF`, which has the following consequences: the controllers are disabled, the PWM outputs go to high-impedance state and a TML interrupt can be generated. The ENABLE input acts like a non-maskable interrupt, which is always executed, no matter of the MotionChip II operation context. As long as this pin is held high, the `AXISON` command has no effect. When the ENABLE input is asserted low MotionChip II operates normally.

4.3. General Purpose digital I/O

The MotionChip II has 19 digital inputs/outputs pins. Some of these I/O pins have also second shared function (see Table 4.1). These pins can be used as I/O only if the shared function is disabled. For example IO#32/HALL1, IO#33/HALL2 and IO#34/HALL3 may be used as general-purpose I/O in motion application configuration where Hall sensors feedback is not required. In a motion application based on a brushless DC motor, these pins will be automatically set as inputs to read Hall signals, when `ENDINIT` TML instruction is executed.

Table 4.1. *MotionChip II general purpose IO*

DSP Name	TML Name	Default State	Function	Description
IOPA2/XINT1	IO#2/LSP	Capture	I/O in	Used as positive Limit Switch
IOPA5/CAP3	IO#5/CAPI	Capture	Enc1Z	Used as Z signal from encoder
IOPB5/T2CMP	IO#13	Out high	I/O out	General purpose I/O
IOPB6/TDIRA	IO#14	Out high	I/O out	General purpose I/O
IOPB7/TCKINA	IO#15	Out high	I/O out	General purpose I/O
IOPD0/XINT2	IO#24/LSN	Capture	I/O in	Used as Negative Limit Switch
IOPE2/PWM8	IO#27	Out high	I/O out	General purpose I/O
IOPE3	IO#28	Out high	I/O out	General purpose I/O
IOPE4	IO#29	Out high	I/O out	General purpose I/O
IOPE5	IO#30	Out high	I/O out	General purpose I/O
IOPE6	IO#31	Out high	I/O out	General purpose I/O
IOPE7	IO#32/Hall3	In	Hall3	Used as third Hall sensor input
IOPF0/CAP5	IO#33/Hall1/ENC2A	In	Hall1	Used as first Hall sensor input
IOPF1/CAP6	IO#34/Hall2/CAPI2	In	Hall2	Used as second Hall sensor input
IOPF2/T3CMP	IO#35/Hall3/ENC2B	In	Hall3	Used as third Hall sensor input
IOPF3/T4CMP	IO#36	In	I/O in	General purpose I/O
IOPF4/TDIRB	IO#37	In	I/O in	Direction input for pulse & direction mode
IOPF5/TCKINB	IO#38	In	I/O in	Pulse input for pulse & direction mode
IOPF6	IO#39	In	I/O in	General purpose I/O

4.4. Analogue Inputs

MotionChip II has 16 A/D channels, divided in:

- 6 A/D channels with predefined functionality, to measure currents, temperature etc;
- 5 A/D channels for setting AXISID needed in multiple axes applications.
- 1 A/D channel allows selecting, at system initialization (at power on or after a reset), the serial communication protocol RS-232 or RS-485. To select the communication protocol RS-232 the voltage applied must be greater then 1.65V; the communication protocol RS-485 is selected when the voltage applied is lower then 1.65V.
- 1 A/D channel to enable/disable, at system initialization, the AUTORUN mode. To enable the AUTORUN mode the voltage applied is lower then 1.65V; the AUTORUN mode is disabled (MotionChip II waits for commands) when the voltage applied is greater then 1.65V.
- 3 A/D channels not used.

The A/D channels functions is described in Table 4.2.

Table 4.2. *MotionChip II analogue inputs*

DSP Name	TML Name	Function	Description
ADCIN0	AD6	–	**Not used**, must be connected to ground
ADCIN1	AD7	TEMP_2	Used to measure drive temperature
ADCIN2	AD2	–	**Not used**, must be connected to ground
ADCIN3	AD5	REF	Used to read analogue reference
ADCIN4	AD4	Vdc	Used to measure DC-link/supply voltage
ADCIN5	AD1	IC	Measure the current through motor phase C
ADCIN6	AD0	IA	Measure the current through motor phase A
ADCIN7	AD3	–	**Not used**, must be connected to ground
ADCIN8	232/485	232/485	Select serial communication protocol RS232 or RS485
ADCIN9	–	Autorun	Enable/Disable Autorun mode
ADCIN10	–	AxisID 4	5^{th} AxisID bit
ADCIN11	–	AxisID 3	4^{th} AxisID bit
ADCIN12	–	AxisID 2	3^{rd} AxisID bit
ADCIN13	–	AxisID 1	2^{nd} AxisID bit
ADCIN14	–	AxisID 0	1^{st} AxisID bit
ADCIN15	–	TEMP_1	Used to measure motor temperature

4.4.1. IA, IC – Motor Currents

For torque/current control, the MotionChip II needs current feedback. In case of three-phase motors, the MotionChip II uses for torque control two currents, the motor current in phase A, read on input ADCIN6, and the motor current in phase C read on input ADCIN5. In case of DC brush motors, the MotionChip II reads the motor current on IA input.

The current sensors may be placed on the motor lines or on the lower legs of the inverter. Figure 4.4 shows how to measure the currents for DC brush or three-phase motors using shunts on the lower legs of the inverter. Current feedback signal polarity relative to motor current is emphasized.

By default, the values measured on IA and IC inputs are interpreted with an offset of half A/D input range. Hence, values higher than mid-point are interpreted as positive currents, while values lower than mid-point are interpreted as negative values. Due to current measurement scheme imperfections, sometimes the A/D mid-point doesn't corresponding exactly to a zero current. For these situations, the MotionChip II can detect automatically the true offset, for each of the A/D inputs. This offset will be considered for current feedback interpretation.

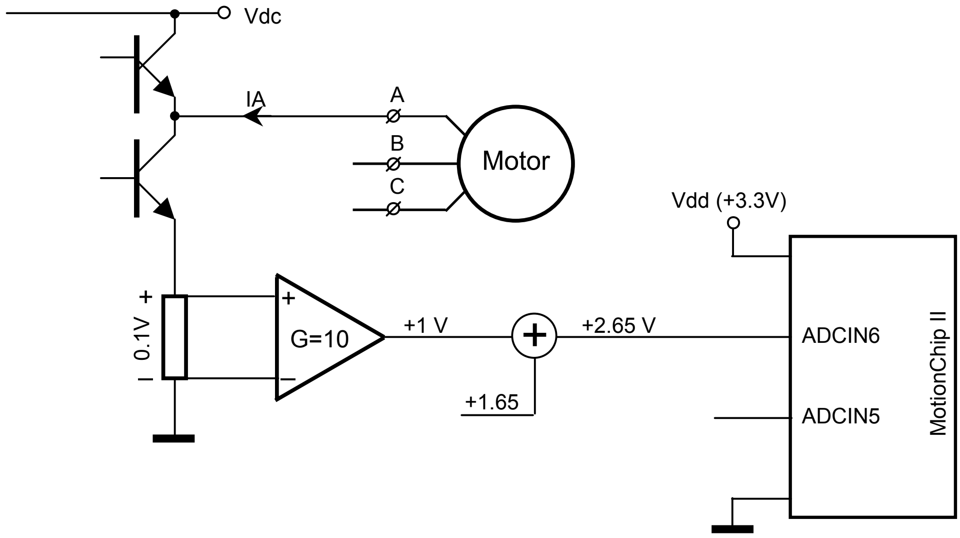


Figure 4.4. *Current measurement from lower-legs of a 2 or 3-phase converter*

4.4.2. VDC – DC-bus Voltage

The MotionChip II has a dedicated analogue input VDC (ADCIN4) for DC-link/supply voltage measure. When this feedback is provided, the following features of the MotionChip II can be enabled:

- Compensation of the DC-bus voltage variation;
- Brake control
- Over-voltage and under-voltage protections

Since DC-link/supply voltage can have only positive values, the value read on VDC input is interpreted without any offset, the DC-link/supply maximum voltage correspond to analogue input maximum voltage (i.e. 3.3V). Figure 4.5 presents an example how to connect the voltage sensor at MotionChip II analogue input.

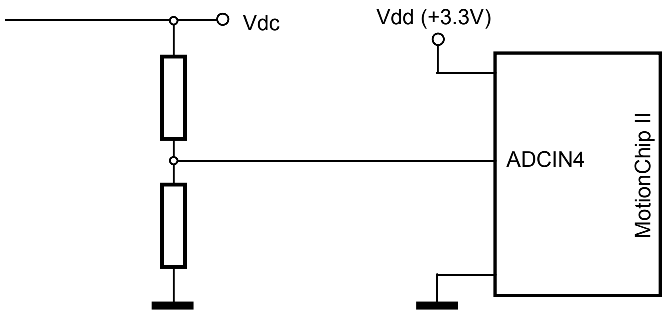


Figure 4.5. *DC-link/supply voltage measurement*

4.4.3. REFERENCE – Analogue Reference

In motion modes with analogue external reference, when internal trajectory generator is disabled, the reference is provided on the analogue input REFERENCE (ADCIN3). For example on this input a potentiometer can be connected to set the speed reference of a motor. By default, the values measured on REFERENCE input are interpreted with an offset of half A/D input range. For applications where the analogue reference can have only positive or only negative values, the default offset can be changed in order to accommodate with these cases.

Figure 4.6 presents an example how to apply an analogue external reference to MotionChip II.

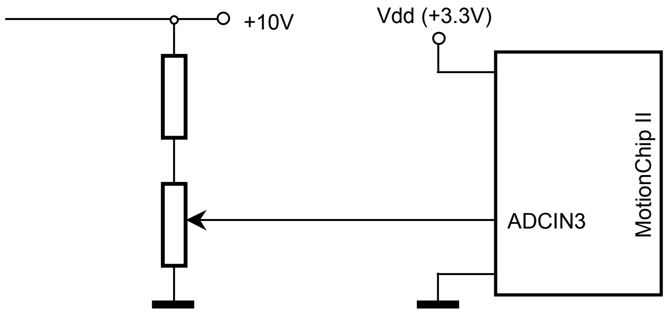


Figure 4.6. *Analogue external reference*

4.4.4. TEMP_1, TEMP_2 – Temperature sensors

The MotionChip II includes two analogue inputs TEMP_1 (ADCIN15) to monitor the motor temperature and TEMP_2 (ADCIN1) to monitor the power stage temperature. The values measured from temperature sensors are used by the software protections. The protections can be programmed to signal when the values read from TEMP_1, TEMP_2 inputs bypass certain programmed limits.

The TEMP_1 and TEMP_2 inputs can have only positive values. Hence, the values read are interpreted without any A/D offset.

Motor temperature sensor can be PTC or NTC. For power stage temperature sensor the user can specify the sensor offset (sensor output at 0°C), which is added to the measured value.

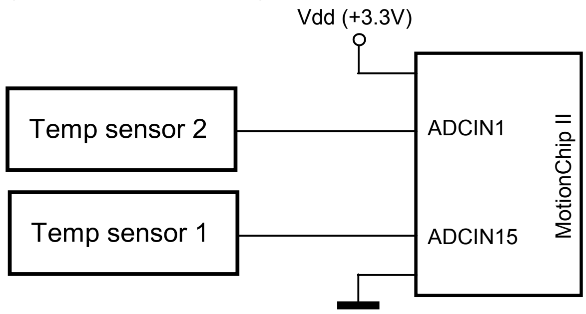


Figure 4.7. *Temperature sensors connected to MotionChip II*

4.4.5. AxisID

In multiple-axis configurations, each MotionChip II based drive needs to be identified through a unique number – the **axis ID**. This is a number between 1 and 255. The axis ID is initially set at power on by reading the MotionChip II analogue input lines ADCIN10 to ADCIN14, as follows:

- Axis ID = 255 if at all analogue inputs ADCIN10 to ADCIN14 the voltage applied is greater then 1.65V;
- Axis ID = 1 to 31, if at least one of the ADCIN10 to ADCIN14 inputs has the applied voltage lower then 1.65V. The axis ID value depends on the analogue inputs combination (see MotionChip II TML Programming User Manual for more information)

Figure 4.8 presents an example how to connect the A/D channels for setting AXISID using a DIP switch.

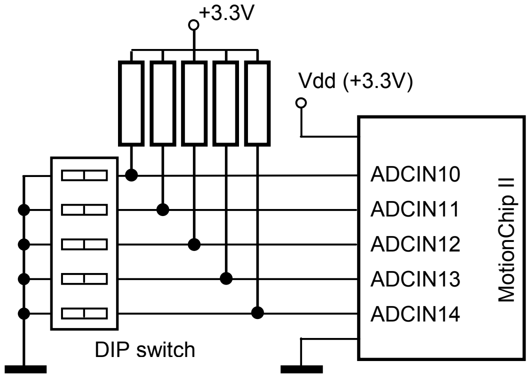


Figure 4.8. *AxisID selection using a DIP switch*

5. MotionChip II Memory

The MotionChip II works with 2 separate address spaces: one for TML programs and the other for data memory. Each space accommodates a total of 64K 16-bit word.

The first 16K of the TML program space (0 to 3FFFh) is reserved and can't be used. The next 16K, from 4000h up to 7FFFh (for 256kbits E^2ROM) is mapped to a serial SPI-connected E^2ROM. This space can be used to store TML programs, cam tables or other user data in a non-volatile memory. The E^2ROM size can be: 64kbits, 128kbits or 256kbits.

The SPI clock rates can be set at 1MHz, 2MHz or 5MHz. The default rate after reset is 1MHz.

Figure 5.18 presents how to interface a serial SPI - E^2ROM with MotionChip II.

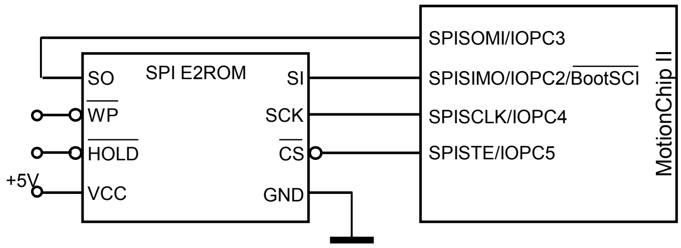


Figure 5.1. *MotionChip II connections with E^2ROM*

The next 2K of the TML program space from 8000h to 87FFh represents the Motion Chip II internal SRAM memory. From it, the first 200h (from 8000h to 81FFh) are reserved for the internal use. The rest from 8200h to 87FFh may be used to temporary store TML programs.

The TMS320LF2407A offers the possibility to connect an external SRAM, which can be mapped in the last 32K more exactly in the address range 8800h to FFFFh (all TML program memory accesses in the address range 0x8000 to 0x87FF are using the internal SRAM). By connecting a 32Kx16 external SRAM, the total TML program space in SRAM memory becomes from 8200h to FFFFh. Figure 5.2 presents the connections with external memory.

The TMS320LF2406A has no external interface, hence only the internal SRAM may be used as TML program memory in the address range 8200h to 87FFh. The remaining TML program memory space from 8800h to FFFFh is invalid.

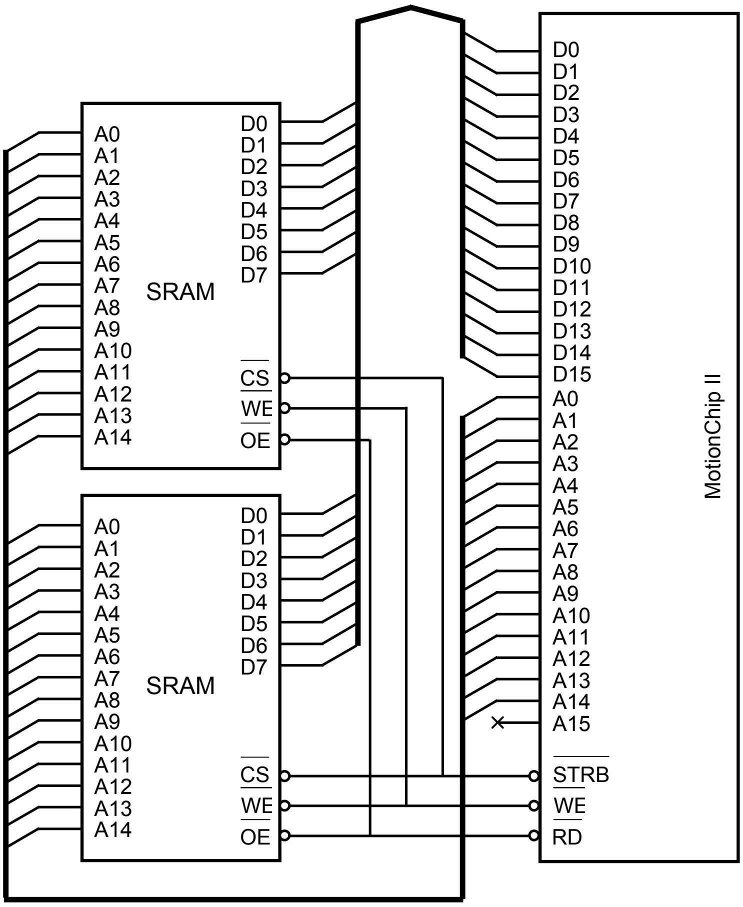


Figure 5.2. *MotionChip II interface with external memory (TMS320LF2407A)*

The data memory space is used to store the TML data (registers, parameters, variables), the cam tables during runtime (after being copied from the E^2ROM memory) and for data acquisitions. The TML data are stored in reserved area, while the others are using the same Motion Chip II internal SRAM memory.

In the data memory space, the internal SRAM is mapped at a different address range 800h to FFFh From this the first 200h, from 800h to 9FFh (corresponding to 8000h to 81FFh in TML program memory space) are reserved for the internal use. The rest from A00h to FFFh corresponding to 8200h to 87FFh in the TML program memory space) may be used for data acquisitions and/or to store cam tables during runtime. As this space is available in both the TML program space and the data space it is the user responsibility to decide how to split it between the two and to avoid overlapping them.

In the case of TMS320LF2407A, if an external SRAM is connected it can be mapped both on the TML program space and in the data space. Typically, the external SRAM is mapped at the same addresses in both the TML program and the data space. Therefore the data memory extends with the external SRAM space from 0x8000 to 0xFFFF.

6. Motion Chip II Communication

6.1. Communication channels

The MotionChip II accepts two types of communication channels:

- Serial RS-232 or RS-485 with on-chip controller
- CAN-bus with on-chip CAN controller

The serial RS-232 communication channel can be used to connect a host with a single MotionChip II based device (see Figure 6.1). The serial RS-485 and the CAN-bus communication channels can be used to create a distributed control network with a host and up to 255 MotionChip II based devices (see Figure 6.2 and Figure 6.3).

When CAN-bus communication is used, any MotionChip II based device from the network may also be connected through RS-232 with a host (see Figure 6.4). In this structure, the axis connected to the host, apart from executing the commands received from host or other axes acts also as a retransmission relay which:

- Receives through RS-232, commands from host for another axis and retransmits them to the destination through CAN-bus
- Receives through CAN-bus data requested by host from another axis and retransmits them to the host through RS-232

This flexibility enables a host to program and monitor a CAN-bus network of devices based on the MotionChip II, using only one RS-232 connection, without the need to have a CAN-bus interface. In this case the CAN-bus protocol is completely transparent for the host.

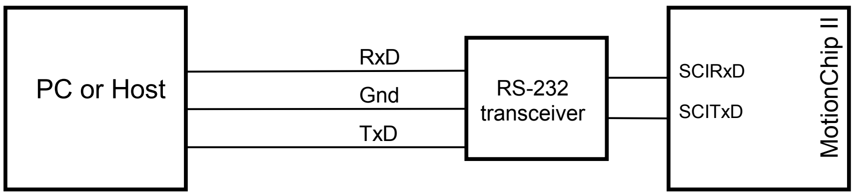


Figure 6.1. *Serial RS-232 communication between a host and the MotionChip II*

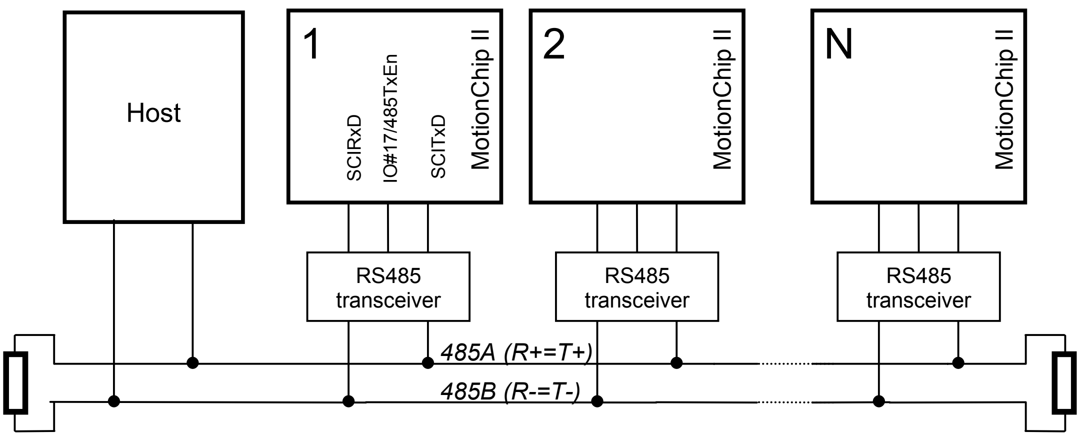


Figure 6.2.*Multi-drop network using serial RS-485 communication*

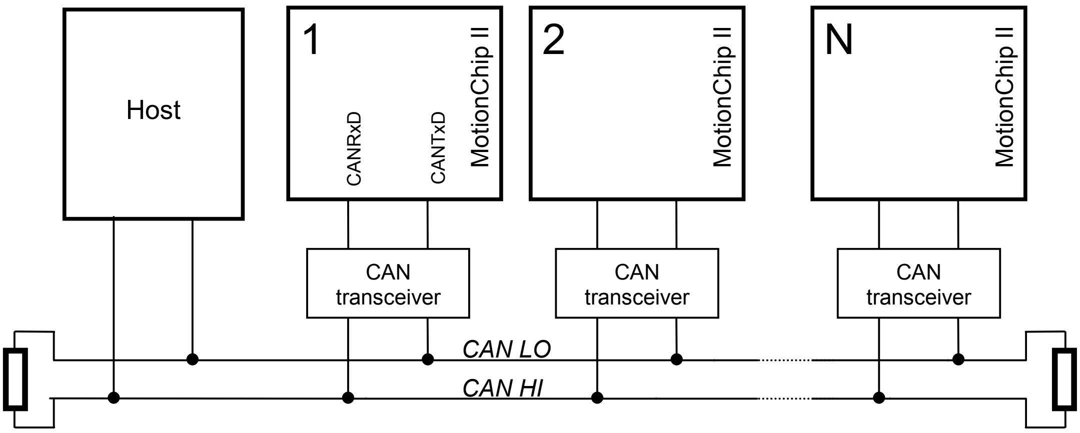


Figure 6.3. *Multi-drop network using CAN-bus communication*

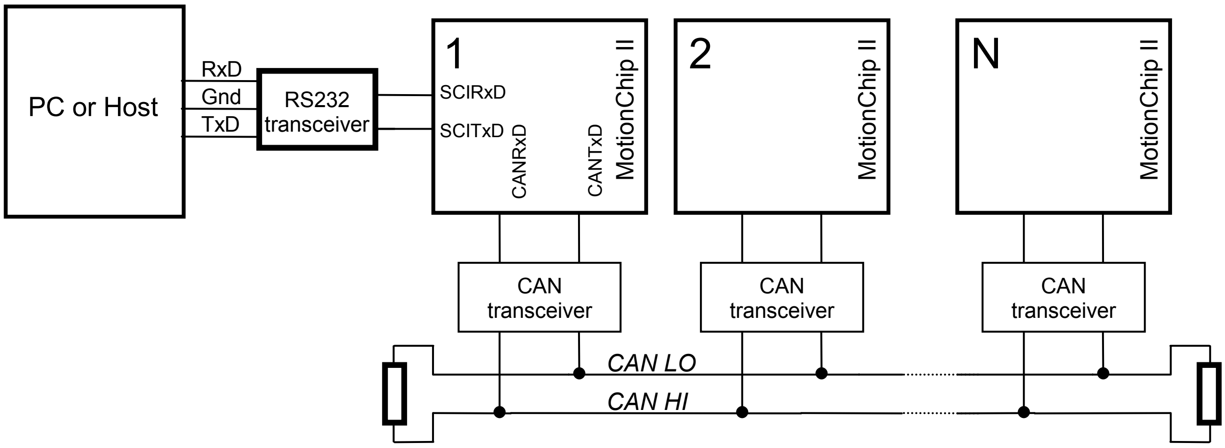


Figure 6.4. *Multi-drop network using CAN-bus communication with host connected through RS-232 to an axis used as communication relay*

6.2. Serial RS-232 and RS485 Interface

The serial communication is based on the on-chip serial communication interface (SCI). This interface provides two pins: SCITxD/IOPA0 for transmission and SCIRxD/IOPA1 for reception. The SCI interface may be used for serial RS-232 or RS-485 communication.

For RS-232, the SCIRxD and SCITxD pins should be directly linked to an RS-232 transceiver (see Figure 6.5).

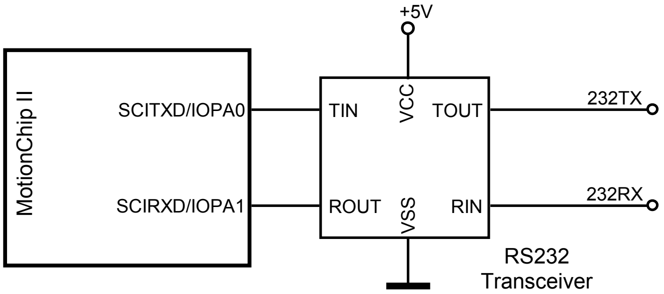


Figure 6.5. *Interface with an RS-232 transceiver*

The communication on RS-485 is enabled when the voltage applied to analogue input ADCIN8 is lower then 1.65V. For RS-485, the two SCI pins and the RS-485 transmission/reception enable pin 485TxEN ($\overline{\text{BIO}}$ /IOPC1), must be connected to an RS-485 transceiver.

The 485TxEN pin is used only when RS-485 communication is selected. The 485TxEN pin is kept low unless asserted high during transmission cycles. Hence, when 485TxEN is asserted low, the reception must be enabled and the transmission disabled. When 485TxEN is asserted high, the transmission must be enabled and the reception disabled.

The Figure 6.6 shows an example how to interface the MotionChip II with an RS-485 transceiver.

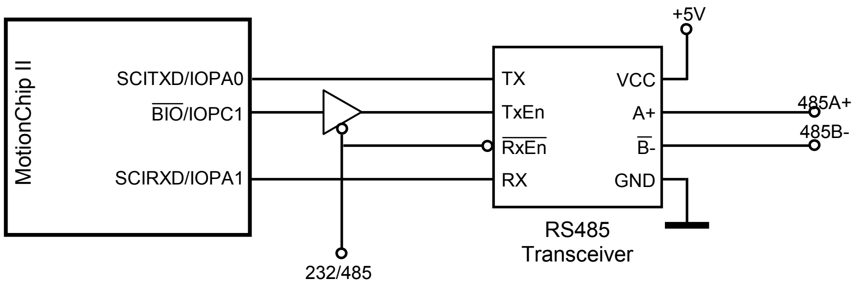


Figure 6.6. *Interface with an RS-485 transceiver*

6.3. CAN-bus Interface

The CAN-bus communication is based on the on-chip CAN controller. The CAN controller requires an external transceiver. Figure 6.7 presents the MotionChip II interface with the CAN transceiver. By default, the CAN-bus baud-rate is set at 500kb.

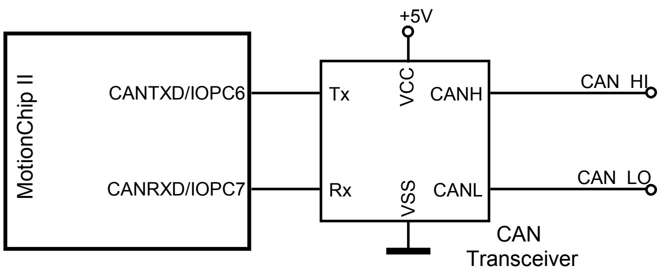


Figure 6.7. *Interface with a CAN transceiver*

7. Development Tools

A complete range of development tools is available for the MotionChip II user. It starts with a basic evaluation kit, the MotionChip™ II Starter Kit (**MCIISK**), accompanied by power module PM-50. On the software side, the powerful **IPM Motion Studio** platform allows you to configure, parameterize and program the motion for your application.

The MotionChip™ II Starter Kit boards contains the MotionChip II, external E^2ROM and RAM memories, an RS-232 interface, and extension connectors, to interface the MotionChip II with external power modules and/or external I/O interfaces. The MCIISK kit can be combined with specific power amplifier modules and motors, thus becoming a complete motion structure for the evaluation of motion applications based on the MotionChip II. See the "**MotionChip™ II Starter Kit User Manual**" for more details about MCIISK kit contents and features.

With the MCIISK board you receive a template for IPM Motion Studio, which is a high level graphical Windows environment for MotionChip II applications development. It allows you to configure and parameterize a motion system (including tuning and auto tuning of controllers) and to define motion sequences using high level integrated tools which automatically generate Technosoft Motion Language source code (TML instructions). Embedded code development tools allow you to further edit or directly compile, link and generate executable code to be downloaded to the MotionChip II. Finally, advanced graphics tools — like data logger, control panel and view/watch of TML parameters, registers and memory — can be used to analyze the behavior of the motion system. See the "**IPM Motion Studio User Manual**" for more details about IPM Motion Studio contents and features.