

Parent	Notes	Function Syntax	Description	Parameters	Returns
Motor.cpp	<p>Recommended object: Motor motor = Motor();</p> <p>The Motor object encapsulates DRV2605 functionality and gives a simpler interface to playing vibration alerts and haptic effects. Only one should be instantiated at a time, as demonstrated in the IntroMode and EngineeringMode sketches</p>	selectMotor(motorID);	Selects the output motor on the grip. Required even if using external actuator is being used - autoCalibrate()	motorID: uint8_t	Void
		isCalibrated();	Checks to see if selected actuator has undergone calibration	None	Boolean Value True = calibrated False = not calibrated
		autoCalibrate();	Runs auto calibration on selected motor. selectMotor() must be called first	None	Void
		playVibAlert(waveform , pwr , onTime , offTime);	Plays vibration alert on selected actuator, LRA is not supported so motorID ≠ 3. Implements a continuous loop, use interrupts or edit function to stop.	waveform: uint8_t pwr: uint8_t onTime: uint8_t offTime: uint8_t	Void
		playFullHaptic(library , effect);	Plays haptic effect from DRV2605 on selected actuator. If LRA is selected (motorID = 3) then library 6 will be used	library: uint8_t effect: uint8_t	Void
		getMotorID();	Gets current motorID	None	Int
		isPlaying();	Checks to see if an actuator is playing a haptic effect	None	Boolean Value True = playing False = not playing
		isPlayingVib();	Checks to see if an actuator is playing a vibration alert. Note 'off times' return a true value. Must be called in interrupt	None	Boolean Value True = playing False = not playing
		stopVibAlert();	Stops the vibration alerting being played. Must be called in interrupt	None	Void
DRV2605.cpp	<p>Recommended object: extern DRV2605 drv2605;</p> <p>The DRV2605 class abstracts away the I2C communications to the DRV2605 chip, providing a streamlined interface for playing effects. Only one DRV2605 object should be instantiated. If using the DRV2605 object from the Motor class use 'extern', as above.</p>	autoCal(ratedVoltage , overdriveClamp , LRA , compensation , backEMF , feedback);	Run the DRV2605's auto-calibration routine on the selected actuator with the values passed in. If auto-calibration is successful (function returns true) the compensation, back EMF and feedback values have been set	ratedVoltage: uint8_t overdriveClamp: uint8_t LRA: boolean compensation: uint8_t* backEMF: uint8_t* feedback: uint8_t*	Boolean value True = successful False = unsuccessful
		playFullHaptic(library , effect , ratedVoltage , overdriveClamp , compensation , backEMF , feedback);	Plays haptic effect from DRV2605 on selected actuator. Pass in compensation, back EMF, and feedback values returned from auto-calibration, or manually adjusted values	library: uint8_t effect: uint8_t ratedVoltage: uint8_t overdriveClamp: uint8_t compensation: uint8_t backEMF: uint8_t feedback: uint8_t	Void
debug.cpp	N/A	freeRAM();	Returns amount of unused RAM space in DRV2605	None	Int

Parameter	Type & Limits	Values	Description
motorID	0 ≤ int ≤ 3	0 = 305-000 1 = 306-109 2 = 308-102 3 = C10-100	Selects the output actuator. Select 3 if using external LRA
waveform	0 ≤ int ≤ 3	0 = Square 1 = Sine 2 = Triangle 3 = Sawtooth	Changes the output waveform for vibration alerts
pwr	0 ≤ int ≤ 100	0% - 100%	The peak output strength of the selected waveform, always goes from 0% to pwr value
onTime	0 ≤ int ≤ 255	Tenths of Seconds	The amount of time the vibration alert vibrates over
offTime	0 ≤ int ≤ 255	Tenths of Seconds	The amount of time between vibration alerts
library	1 ≤ int ≤ 6	1 - 5 = ERM 6 = LRA	The haptic library on the DRV2605 to be used
effect	1 ≤ int ≤ 123	See Effect ID table	The haptic effect on the DRV2605 to be used
ratedVoltage	0 ≤ int ≤ 255	Voltage applied to ERM = ratedVoltage x (5.44 V / 255)	See full DRV2605 datasheet for detailed explanation and calculation of LRA voltage
overdriveClamp	0 ≤ int ≤ 255	Voltage applied to ERM = overdriveClamp x (5.6 V / 255)	Peak voltage allowed in all modes of DRV2605 operation
LRA	bool	True = LRA Mode, False = ERM Mode	Sets bit 7 of register 0x1A, putting the DRV2605 into LRA or ERM mode
compensation	0 ≤ int ≤ 255	Auto Calibration Compensation Coefficient = 1 + compensation / 255	Manually adjusts compensation for resistive losses in the driver
backEMF	0 ≤ int ≤ 255	Auto Calibration Back EMF (V) = (backEMF / 255) x (4.88 V / BEMFGain)	Manually adjusts results for back EMF of the actuator, BEMFGain set by feedback
feedback	int, only specific values valid	N/A	Manually adjusts feedback control register, see full DRV2605 datasheet for details

I2C Addresses

Using 7-bit Addressing

Capacitive
Touch Driver
MPR121

0x5D

Haptic Driver
DRV2605

0x5A

Pin Mapping

