**Functions drive\_commands.h**

|  |  |
| --- | --- |
| **Syntax** | void driveStop() |
| **Purpose** | stops right, left and straff drive motors |
| **Entry Conditions** | NONE |
| **Code**  **Example** | //engage the drive  driveGo(127);  //wait for 1 second  wait1Msec(1000);  //stop the drive  driveStop(); |

|  |  |
| --- | --- |
| **Syntax** | void driveGo(int left, int right, int straff) |
| **Purpose** | engage the right, left and straff drive motors at the desired velocity. |
| **Entry Conditions** | * *left* - represents the desired velocity of the left drive system * *right* - represents the desired velocity of the right drive system * *straff* - represents the desired velocity of the straff drive system |
| **Code**  **Example** | //drive at a 45 degree angle  driveGo(127, 127, 127);  //wait for 1 second  wait1Msec(1000);  //stop the drive  driveStop(); |

|  |  |
| --- | --- |
| **Syntax** | **void** driveGo(**int** left, **int** right) |
| **Purpose** | engage the right and left drive motors at the desired velocity. |
| **Entry Conditions** | * *left* - represents the desired velocity of the left drive system * *right* - represents the desired velocity of the right drive system |
| **Code**  **Example** | //drive forward  driveGo(127, 127);  //wait for 1 second  wait1Msec(1000);  //stop the drive  driveStop(); |

|  |  |
| --- | --- |
| **Syntax** | void driveGo(int velocity) |
| **Purpose** | engage the right and left drive motors at the desired velocity. |
| **Entry Conditions** | * *velocity* - represents the desired velocity of the left and right drive system |
| **Code**  **Example** | //drive forward  driveGo(127);  //wait for 1 second  wait1Msec(1000);  //stop the drive  driveStop(); |

|  |  |
| --- | --- |
| **Syntax** | void driveFor(int time, int left, int right, int straff) |
| **Purpose** | engage the right, left and straff drive motors at a desired velocity, for a set period of time in milliseconds. |
| **Entry Conditions** | * *time* - the amount of time the drive will be active in milliseconds * *left* - represents the desired velocity of the left drive system * *right* - represents the desired velocity of the right drive system * *straff* - represents the desired velocity of the straff drive system |
| **Code**  **Example** | //drive at a 45 degree angle for 2 seconds  driveFor(2000, 127, 127, 127); |

|  |  |
| --- | --- |
| **Syntax** | void driveFor(int time, int left, int right) |
| **Purpose** | engage the right and left drive motors at a desired velocity, for a set period of time in milliseconds. |
| **Entry Conditions** | * *time* - the amount of time the drive will be active in milliseconds * *left* - represents the desired velocity of the left drive system * *right* - represents the desired velocity of the right drive system |
| **Code**  **Example** | //drive forward for 2 seconds  driveFor(2000, 127, 127); |

|  |  |
| --- | --- |
| **Syntax** | void driveFor(int time, int velocity) |
| **Purpose** | engage the right and left drive motors at a desired velocity, for a set period of time in milliseconds. |
| **Entry Conditions** | * *time* - the amount of time the drive will be active in milliseconds * *velocity* - represents the desired velocity of the left and right drive system |
| **Code**  **Example** | //drive forward for 2 seconds  driveFor(2000, 127); |

|  |  |
| --- | --- |
| **Syntax** | void driveDistance(int left, int right, int straff) |
| **Purpose** | engage the right, left and straff drive motors for a desired distance |
| **Entry Conditions** | * *left* - the desired distance the left drive system must travel * *right* - the desired distance the right drive system must travel * *straff* - the desired distance the straff drive system must travel |
| **Code**  **Example** | //travel 1414 ticks at 45 degrees  driveDistance(1000, 1000, 1000); |

|  |  |
| --- | --- |
| **Syntax** | void driveDistance(int left, int right) |
| **Purpose** | engage the right and left drive motors for a desired distance |
| **Entry Conditions** | * *left* - the desired distance the left drive system must travel * *right* - the desired distance the right drive system must travel |
| **Code**  **Example** | //travel forward 1000 ticks  driveDistance(1000, 1000); |

|  |  |
| --- | --- |
| **Syntax** | void driveDistance(int distance) |
| **Purpose** | engage the right and left drive motors for a desired distance |
| **Entry Conditions** | * *left* - the desired distance the left and right drive system must travel |
| **Code**  **Example** | //travel forward 1000 ticks  driveDistance(1000); |

|  |  |
| --- | --- |
| **Syntax** | void xStraffGo(int x, int y, int twist) |
| **Purpose** | the robot x-drive straff is activated |
| **Entry Conditions** | * *x* - the horizontal velocity vector * *y* - the vertical velocity vector * *twist* - the rotate velocity vector |
| **Code**  **Example** | //engage straffing  xStraffGo(90, 90, 0); |

|  |  |
| --- | --- |
| **Syntax** | void xStraffGo(int x, int y, int twist) |
| **Purpose** | the robot x-drive straff is activated |
| **Entry Conditions** | * *x* - the horizontal velocity vector * *y* - the vertical velocity vector |
| **Code**  **Example** | //engage straffing  xStraffGo(90, 90); |

|  |  |
| --- | --- |
| **Syntax** | void xStraffFor(int time, int x, int y, int twist) |
| **Purpose** | the robot x-drive straff is activated for a desired amount of time |
| **Entry Conditions** | * *time* - the amount of time in milliseconds * *x* - the horizontal velocity vector * *y* - the vertical velocity vector * *twist* - the rotate velocity vector |
| **Code**  **Example** | //engage straffing for one second  xStraffFor(100, 127, 127, 90); |

|  |  |
| --- | --- |
| **Syntax** | void xStraffFor(int time, int x, int y) |
| **Purpose** | the robot x-drive straff is activated for a desired amount of time |
| **Entry Conditions** | * *time* - the amount of time in milliseconds * *x* - the horizontal velocity vector * *y* - the vertical velocity vector |
| **Code**  **Example** | //engage straffing for one second  xStraffFor(100, 127, 127); |

|  |  |
| --- | --- |
| **Syntax** | void xStraffDistance(int x, int y, int twist) |
| **Purpose** | the robot x-drive straff is activated for a desired distance |
| **Entry Conditions** | * *x* - the horizontal distance vector * *y* - the vertical distance vector * *twist* - the rotate distance vector |
| **Code**  **Example** | //engage straffing at a 45 degree angle  xStraffDistance(90, 90, 0); |

|  |  |
| --- | --- |
| **Syntax** | void xStraffDistance(int x, int y) |
| **Purpose** | the robot x-drive straff is activated for a desired distance |
| **Entry Conditions** | * *x* - the horizontal distance vector * *y* - the vertical distance vector |
| **Code**  **Example** | //engage straffing at a 45 degree angle  xStraffDistance(90, 90); |

|  |  |
| --- | --- |
| **Syntax** | void turn(int degrees) |
| **Purpose** | allows robot to turn a specific amount of degrees |
| **Entry Conditions** | * *degrees* - the desired amount of degrees for the robot to turn. Negative turns left, positive turns right. |
| **Code**  **Example** | //turn right 90 degrees  turn(90); |

|  |  |
| --- | --- |
| **Syntax** | void flipDrive() |
| **Purpose** | switches around the left and right drive and reverses the straff drive. |
| **Entry Conditions** | NONE |
| **Code**  **Example** | //Flip right and left for mirrored autonomous  flipDrive(); |