View of modules from the top:

1	**2**
3	**4**

Calculating the angles for each driver:

Assume Point A ,with coordinates (x, y), is the location to which the robot needs to move, with the x-axis being the North-South axis of the robot and the y axis being the East-West axis of the robot (the coordinate (0,0) is at the centre of the robot).

Direction of A relative to Module 1 = tan-1(y/x)

Direction of A from Module 1 relative to compass north = tan-1(y/x) + imu_north

Repeat the same for the other modules while changing the signs for each module based on their quadrant:

Module 2: tan-1(y/-x) + imu_north

Module 3: tan-1(-y/-x) + imu_north

Module 4: tan-1(-y/x) + imu_north

Calculating speed of M3058 motor:

To calculate the speed of the motor in RPM we can use the formula:

RPM = (Angular velocity x 60 x gear ratio) / 2π

Where angular velocity = linear velocity/wheel radius

Hence the additional parameters we need will be the linear velocity, wheel radius and gear ratio.

Code:

The swerve_turn function requires 3 sets of parameters as input for each module. The chassis_control_task function will calculate and store all the parameters in a structure to be accessed when calling the swerve_turn function for each module.

```c
void chassis_control_task(float x, float y, float linear linear_velocity, float wheel_radius, float imu_north, float gear_ratio){
    float angular_velocity = linear_velocity / wheel_radius;
    RPM = (int16_t)((angular_velocity * 60 * gear_ratio)/(2 * M_PI));

    struct RobotParameters {
    uint8_t module_num[4];
    float gm6020_ang[4];
    float angular_velocity;
    int16_t m3058_rpm[4];
    } robot_params = {
        .module_num = {1, 2, 3, 4},
        .gm6020_ang = {atan2(y, x) + imu_north, atan2(y, -x) + imu_north, atan2(-y, -x) + imu_north, atan2(-y, x) + imu_north},
        .m3058_rpm = {RPM, RPM, RPM, RPM}
    };

    for(long i = 0; i<4; i++){
        swerve_turn(robot_params.module_num[i], robot_params.gm6020_ang[i], robot_params.m3058_rpm[i]);
    }

}
```