

# GSOC 2017 : Proposal to Ardupilot

Project : Make APMRover2 a reference of generic UGV platform

## About Me

### ***Basics Information***

Name: KANCIR Pierre

Address :

Email :

Telephone :

Skype :

Time Zone : Paris, France UTC+1

LinkedIn : [https://www.linkedin.com/in/pierre-kancir/?locale=en\\_US](https://www.linkedin.com/in/pierre-kancir/?locale=en_US)

Source Control Username : <https://github.com/khancyr>

Youtube channels :  
<https://www.youtube.com/playlist?list=PL6sCNLbHuYxYyFWLTq3E-R2cV1CrEoYOe> and  
<https://www.youtube.com/playlist?list=PL6sCNLbHuYxaLtlzADD5zHxixUtQPSvk>

### ***Education :***

University : Université de Bretagne Sud, France

Degree : PhD

Current Year : 4

Graduation : 2017

PhD subject : Mechanisms for optimized information sharing and synchronization between embedded systems within a swarm of drones.

## Project information :

### ***Proposal title :***

Make APMRover2 a reference of generic UGV (Unmanned ground vehicle) platform

## ***Proposal abstract :***

Current drones require a reliable and proven autopilot in order to accomplish autonomous missions. The development of a complete autopilot requires advanced knowledge in electronics and software in addition to knowledge on the robot's type dynamic (multirotor, plane, rover, etc.). This is generally a long and expensive development, which is why it is often preferable to use an existing solution. The Ardupilot platform has the advantage of being low cost, open source with a strong community and industrial partners, and is usable on several types of platforms. Indeed, this autopilot is usable on UAV, UGV, plane, boat, ROV etc. This platform is widely used worldwide and has proved its effectiveness on several occasions: Ardupilot Rover won the 2013 and 2014 sessions of the Sparkfun Autonomous Vehicle Competition and Ardupilot : Plane won the 2014 and 2016 session of the Outback Challenge in Australia. Originally linked to a hardware platform, its hardware support has been widely extended with its use in Linux with SBCs like the Raspberry Pi.

Nevertheless, my academic works show that if ROS starts to be commonly used in robotics research and development, no high capability reference robotic platform exists at accessible price (i.e. < 10000€). This is a major problem as scientists spend either a lot of time to develop a platform or lot of money to customize an already expensive platform. On my PhD MRS (Multi Robots System) project, for example, instead of focussing on MRS problems a large amount of time was spent on building robots, which was not the real goal of the research project. I believe that Ardupilot can be part of a solution to reduce build-time by providing a generic and proven autopilot for all vehicle types, in the same way as ROS reduced software development time for many robotics projects!

The ardupilot features are already numerous and I have already worked on helping ardupilot to have good compatibility with ROS. Ardupilot rover version (APMRover2) is well behind Arduplane and ArduCopter in capabilities as they attract more attention from developers. One obvious example is that Rover has only a very basic avoidance system (turn 45° on obstacle, no vision or lidar system). It is also missing some autonomous features (on spot turn etc.), and more significantly is missing a proper motor configuration abstraction layer. While ArduCopter is capable of being configured in 4,6,8 motors copter with brushed (PWM) or brushless motor (RC PWM), the rover version is still limited to two modes: rover with propulsion and a steering servo (normal mode) and tank mode (skid steering mode), with only support on brushless motor (RC PWM). Those modes assume only two motors outputs are used. Respectively, throttle and steering servo and right motor and left motor.

My proposal to this GSOC is split in two parts. Firstly, reduce the gap between the APMRover2 and the other vehicles to provide a good generic UGV platform capable of multiple configurations, with advance feature such as avoidance. Secondly, provide a good set of tutorials on the usage of ROS and ardupilot either for single unit control or on simulated unit on Gazebo simulator!

## ***Features and Deliverable Specifications :***

- Task 1 : Create a new library or rework existing one for motor control to permit the usage of brushed motors. The use of brushed motor is different than for brushless motor. In those, the PWM signal sent to ESC (Electronic Speed Controller) is constrained between 1 and 2ms at 20Hz. Value greater than 1.5ms are for forward direction, and under 1.5ms for reverse direction. 1.5ms is the neutral. On brushed

motor, the motor is generally driven by a H-Bridge that will handle the power given to the motor. The H-Bridge expect a PWM signal at certain rate. This PWM signal transmit the power information with its duty cycle *"The term duty cycle describes the proportion of 'on' time to the regular interval or 'period' of time; a low duty cycle corresponds to low power, because the power is off for most of the time. Duty cycle is expressed in percent, 100% being fully on."* [WIKIPEDIA](#). If the power can be controlled, the direction can't; an extra signal is generally sent to the H-Bridge to control the motor direction. The new library should take this into account.

- Task 2 : Create a new abstraction layer for Rover to provide frame configurations other than the two existing ones and simplify their interdependence. In fact, the recent addition of reverse drive feature on APMRover2 normal mode isn't working on the tank mode.
- Task 3 : Modify APMRover2 capabilities to include avoidance works from copter.
- Task 4 : Create documentation and tutorial on how to use Rover and Rover + ROS combination.
- Task 5 : Create a new set of autotest on Rover to provide reliable testing on its features.

## ***Project Timeline and List of Deliverables:***

May 30 - 2 June :

- Planning and specifications writing with mentor.
- Building of the test bed and rover.
- Blog post about the project and document the rover building.

5 - 9 June :

- Creation of the initial new motor lib. The difficulty will be to design a simple API and to rework the current motor handling that is split between multiple files and functions.
- Test and correct for regressions on old controllers: test on SITL (Ardupilot Software In The Loop simulation) and real rover.
- Submit the code to the main repo.
- Dedicate some time for community helping.

12 -15 June :

- Implement the brushed motor controller. Copter already implements the brushed motor support. There shouldn't be difficulties here. I have already tried to implement this (<https://github.com/khancyr/ardupilot/commits/pr-brushed-motors>) but it needs the work from week 1 and a way to handle direction signal.
- Test the brushed motor controller
- Submit the code to the main repo.
- Correct the code from week 1 on community review.
- Dedicate some time for community helping.

19 - 23 June :

- Complete the previous works and tests if unfinished.
- Correct the previous committed works on community review.
- Write the documentation and tutorials. Documentation should include wiring diagram and explanation of the new solution.
- Make a video and blog post about progress.

26 - 30 June : Evaluation time and buffer week

- Finish any leftover work and make last corrections.
- Prepare evaluation.
- Bonus: create special RC PWM mode : Oneshot or implement a way to chose PWM frequency.
- Planning, specification of tasks 2 and start of task 2.

3 - 7 July :

- Creation of the initial new control lib. The current implementation uses plenty of “if statement” in the various file that handle the difference between normal and tank mode. The work on the new control lib could be an opportunity to switch to something like the Frame\_class in copter and use the rover mode split as in <https://github.com/peterbarker/ardupilot/tree/rover-modes> for easier mode and control management. This will simplify the tracking of which functionalities are working in a mode or not and the addition of new ones.
- Test and correction for regressions on old controllers
- Submit the code to the main repo
- Dedicate some time for community support.

10 - 14 July :

- Correct the code from previous week on community review
- Complete the previous works and tests if unfinished
- Implement a new mode for scientist: Ackerman steering with rover parameterization or 4 wheel drive. These are the main modes generally used by scientists, so providing an initial implementation inside APMRover2 should attract them.
- Test of the new steering modes.
- Submit the code to the main repo.
- Dedicate some time for community support.

17 - 21 July :

- Complete the previous work and tests if unfinished
- Correct the previous committed works on community review
- Write the documentation and tutorials
- Make a video and blog post about progress

24 - 28 July : Evaluation time and buffer week

- Finish the leftover and last corrections
- Prepare evaluation
- Start of task 3

### 31 - 4 August : Tasks 3

- Adding of avoidance on rover : allow input from all sensors, switch to AP\_Avoidance instead of 45° turn.
- Make correction and improvements in avoidance if needed.
- Test of the implementation with high risk process as in <https://www.youtube.com/watch?v=QJQOFS4Ylfw> . But with a 20cm rover ...so my teddy bear will take risks for me.
- Submit the code to the main repo.
- Write the documentation and tutorials.
- Make a video and blog post about progress.

### 7 - 11 August : Tasks 4

- Complete the previous work and tests if unfinished.
- Correct the previous committed works on community review.
- Write the documentation and tutorials about ROS usage with APMRover2.
- Test avoidance with ROS. If everything is working, the rover should provide sensors information to ROS that will decide of how to react for avoidance. Or do SLAM.
- Test rover improvements with other ardupilot GSOC project (e.g. safe return to launch)

### 14 - 18 August : Task 5

- Write more documentation and tutorials about ROS from community request.
- Write a complete autotest suite for rover that takes into account all its capabilities. The current autotest suite is very basic (see <https://github.com/ArduPilot/ardupilot/pull/5409>), and is missing plenty of features. Implementing a test suite (as in the unfinished work <https://github.com/ArduPilot/ardupilot/pull/5766>) will greatly improve the APMRover2 and ease addition of further capabilities.

### 21 - 29 August : Final week and evaluation

- Finish everything - especially documentation! Even if I am confident in my programming skills when we work with hardware, anything can happen!

Required hardware : I already have a Pixhawk 1 controller for Ardupilot. However, I would need a rover base for real testing and at least a rangefinder for avoidance. Ideally 2 rovers : a brushless rover with servo steering and a brushed tank mode rover. That way I can test both existing configurations with both motor type. Provisional budget : 250 - 300 € maximum, as the rovers just need to be big enough to drive with a Pixhawk, GPS, rangefinder and a small battery. This will be discussed during community bonding period

with mentor to have time to receive the materials for week 1 or 2 on worst case. I lack artistic sense but I have a reflex camera that will help to shoot good tutorial video.

## ***Summer Availability / Additional notes***

My academic works will end in may 2017 and I resigned from my work for professional and ethical reasons, so I will be full time on GSOC at around 39h per week since my wife will try to kill me if I continue to work when she come back from her work ... I don't intend to work much on week-end as believe that resting and family are important both for health and better commitment to work. But I will use some time in week-end to do some real-world testing and try explain to Rennes city Fablab people how an autonomous rover works. I will use some materials at Rennes Fablab - for example, an oscilloscope to check PWM signals (I don't own one).

## ***Related works and contributions***

My PhD was done in a private startup company of 5 persons specialized in robotics. As I was the only engineer in software and electronic, I undertook plenty of robotics project on my own (some of which involved Ardupilot).

<https://www.youtube.com/watch?v=BeOEsEA0JYA>

<https://www.youtube.com/watch?v=e6VII-EMrEI> : Drone with wired connection and intelligent winch

As part of my PhD, I also contributed to various ROS package, Gazebo and Ardupilot (<https://github.com/ArduPilot/ardupilot/commits?author=khancyr>). My previous contributions to Ardupilot are mostly related to codebase cleanup and upgrade to C++11, improvements to SITL, some bug corrections, and various small capabilities addition to rover. See my youtube channel for some example.

## ***Sum up***

I believe that my proposal is an opportunity to bring more developers to Ardupilot project and even better scientists! The rover platform is lacking some essential features that could make it a good generic mobile robot platform, and it is a pity as a rover is less dangerous and easier to use than a copter for everybody! With my PhD experience on swarm, I have the knowledge both in scientist needs and in what a good mobile platform must be, and in addition of my programming skills that make me a good candidate for this GSOC. Moreover, my past contributions to Ardupilot and my experience with ROS give me an incredible opportunity to share my knowledge to a great community. Indeed, some developers are currently needing advice on these subjects to make more advances robots with ROS avoidance or vision system. Even if I don't master these topics, I know how to setup Ardupilot and ROS to use them and I hope to be able to demonstrate these capabilities with a new rover that will bring Ardupilot more popularity as a reference open source platform for mobile robotic!