

**QAG DATA BUS**

**Design Specifications**

tmassey

QuadAutomotive Group

9/25/2012

Table of Contents

[Introduction 2](#_Toc336735437)

[Theory 1: Push Only Operation 2](#_Toc336735438)

[Theory 2: Pull Only Operation 3](#_Toc336735439)

[Theory 3: Push/Pull 4](#_Toc336735440)

# Introduction

It is the aim if this document to describe the specifications and operation of the QAG 16 bit Data BUS. The purpose of the data bus is to provide a high speed parallel bus for communications between each of the Processors. It is a way to provide threaded processing through the use of independent specialty applications running on their own processors. The basis for everything is through a Bus Control Processor. The Bus control processor controls the timing of the bus and who has access when. However I have three theories on the operation of the bus. The first is a push only mechanism. The second is a pull only operation and the third being a push pull mechanism. The schematic allows all three to operate however the key is in the bus drivers in the firmware and the Control Processors firmware.

# Control Bus Operation

The concept is based heavily on the way that Parallax’s Propeller Chip works. The Propeller is a multi-core Microcontroller and uses what they call a hub to give each of the cores a time slice of the shared memory and pins. It operates with a base clock and the hub clock operates much like the R/W clock, and address clocks shown in Figure 1. The Hub clock operates at half the speed of the base clock. Now in my theory there are 2 chip address pins and each of them tells which chip has access to the bus. All of the operation modes actually will operate on this concept. They each get a clock pulse for writing and then a clock pulse where the data is latched. In addition there are 4 pins on the control Process that indicate if a processor is alive and have data to push or pull. Each processor then sets it’s pin high to indicate that on the next trip around to get added into the queue for sending data. This would speed up the bus since we have addresses for 4 chips but only use three at this point in time. So if the Control Process got to a processors address or time slot and the address pin for that chip was Low or false the control processor would move on to the next processor address. In addition there is an address/data clock this clock indicates that the current bytes on the bus are either an address or a value. To keep everyone in sync we toggle this every two 2 clock cycles. So we write data on the first clock high we read on low. The

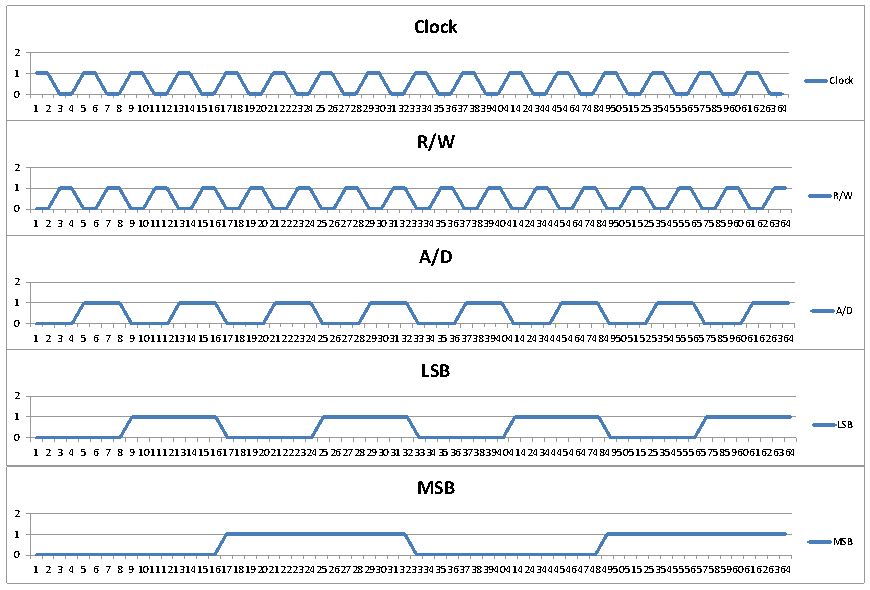


Figure 1

# Push/Pull Processor Control

Each chip is given a range of register addresses. And if their address is put on the bus they respond with the value. The cool thing about this method is a processor can push one of it’s addresses on to the stack and then know that it also needs to respond with the value. However the Flight controller will need to be aware of those registers and know that it needs to react to the change in that register. So here is how I see this working. The IMU is not going to push out the sensor data it will wait for requests to it’s values, unless it detects a stability issue. In that for instance a Free Fall is detected in the accelerometer. That condition would then need to push that to the top of the stack or buffer for it’s next cycle. Otherwise the Flight controller will request the data it wants for stable flight. The same goes with Navigation. The only time it would issue updates to the bus would be when an update is needed. If the craft is heading in the right direction the navigation is simply going to keep on requesting as needed updated Heading, Latitude, Longitude, and altitude from the IMU. The Flight controller will request actual roll, pitch, and yaw as well from the IMU.

# Example Operation

So lets look at an example of this in operation. The vehicle is on the ground throttle is set to 0 or disarmed. The IMU has no requests, the Flight Controller has nothing to request . The Navigation Gets a command from the Communications Unit to take off and ascend to 50 Ft above ground. The navigation computer will first make a IMU request for current altitude. The IMU will respond with the data for that register. The Navigation Computer will store that as the Ground Altitude, and Current Altitude. The Navigation computer will send a 0Yaw value. The flight controller will store that value. The Navigation computer will send a 0 Pitch value and the Flight Controller will store that value. The Navigation Computer will send a 0 roll value and the flight controller will store that value. The navigation computer will send a throttle value of 25% and the flight controller will store the value. However at this point the flight controller now has something to do. Since the vehicle was on the ground the main loop in the Flight Controller was doing nothing since the base throttle level was 0 however now it is 25% of full throttle and it will spin up the Motors. Now the loop needs to start request roll, pitch, and Yaw rates from the IMU. And Control the Aircraft. The same is true with the navigation computer now it need to keep track of altitude. So as the Aircraft climbs the Navigation will keep asking for the Altitude and the IMU responds. The FC keeps the craft stable by requesting the calculated roll pitch and yaw form the IMU. When the Navigation computer sees that the vehicle has exceeded the altitude commanded it sends a Throttle down of 1%. If the vehicle doesn’t descend after a certain period it sends another command to throttle down. At the same time the FC is updating it’s base throttle and slowing down the Motors. This keeps going on to keep the vehicle stable. So the Vehicle now drops below the specified altitude and we need a command to up the throttle. Then the vehicle receives a RF command to turn to a heading of 25o . The navigation computer now needs to figure out which way it is pointing. So it will request current heading. Nothing has changed in the altitude so both the FC and the NAV computer keep on doing their normal thing while the Nav computer calculates that it needs to Yaw 25o to the right. So it sends a Yaw rate of +10. At this point the FC saves that and applies it to it’s offsets. It also keeps on requesting IMU data and now the NAV computer is requesting both altitude and Heading from the IMU. Once the Vehicle reaches it heading the NAV computer sends a yaw rate of 0 to stop it from spinning to the right. And keeps on tracking heading and altitude. Now we get crazy and the NAV computer gets a new waypoint added via RF communications. No big deal it stores the waypoint and keeps on keeping track of altitude and heading. Sending commands hear and there to keep the FC on track based on throttle and roll, pitch, yaw. But then we get a command to make way to the waypoint. So now the Nav Computer must first figure out where it is so it requests Lat and Long values from the IMU. The NAV computer decides that the heading is fine all the while basic nav stability operations are taking place. Then the nav computer calculates that in order to move to that position we need to set a 5% forward pitch, and increase base throttle by 10%. So the Nav computer send the set pitch register and throttle increase. Since the Navigate command also sent along with it the speed of travel the NAV computer is now requesting all of the registers it needs keeping track and updating where necessary the throttle, roll, pitch, and yaw until it reaches its destination. Then it will baseline the throttle, and pitch to begin hover operations again. All the while the FC has been requesting the sensor roll, Pitch and yaw rates to keep the vehicle stable. Then a land command comes and the NAV computer sends a gradual decrease to throttle until the vehicle is on the ground however it also tries to keep the descent to a slow stable and smooth rate. Once on the ground the NAV Computer will notice through a request for the altitude and the lack of change tells it that it is on the ground as well a Distance sensor will also indicate that it is safely on the ground and the NAV Computer will set throttle to 0.