Go-Baby-Go Universal Control

Sponsor Information:	**Dr. Sarah Oman,** Mechanical Engineering Sarah.Oman@nau.edu **Dr. Eck Doerry,** SICCS Eck.Doerry@nau.edu

Project Description:

Children with limited mobility often do not receive the much needed exposure to socialization to appropriately cognitively develop. Existing research shows that enabling young children with self-control of their own environment can have meaningful impacts on the long term outcomes given such impairments as cerebral palsy or muscular dystrophy.

The Go Baby Go (GBG) project that started at the University of Delaware has developed a set of DIY cars for families with children with mobility restrictions. These cars have been designed on commercially available ride on toy car platforms (like Power Wheels) and have been deployed worldwide by the GBG team. These cars have shown to be a cost-effective means of enabling young children to move and interact with their peers, but they are also quite limited, simply due to the fact that retrofits are specifically to be cheap and DIY for parents, i.e., a few common Radio Shack parts, and minimal-tools modifications. In fact, the main modifications are just a big red spring-loaded "go" switch in the center of the steering wheel, and an easy-access kill switch for parents. This leads to the following issues:

- There is no progressive speed control. It's on-off only, leading to jerky behavior that is frightening to many kids. Reverse gear is also not accessible to many mobility-limited kids.
- The steering is unmodified, meaning there is almost no steering for those with upper body impairments; they can only lean on the wheel and "go" switch.

The goal of this project is to prototype the next major version of the GBG project kit, *specifically to design a universal control for children that extremely limited mobility of their arms and/or legs*. The project centers around developing modern microprocessor control of steering and the traction motor. Along with endless "smart" control fidelity, this will allow unlimited control interface possibilities, both for the car's young users, and remotely by parents.

Obviously, this system has both mechanical and computational elements, so the project will be tackled by independent but loosely connected Capstone teams from Mechanical Engineering and Computer Science. The projects will be independent because each team will focus on their expertise area without having to tightly collaborate with the other team, by simply "simulating" the other side's part. For instance, the CS team will develop the control interfaces and logic…but these will control a *simulated* car. Similarly, the ME team will focus on building the steering and propulsion elements, and will prove them with just rudimentary analog controls. The projects are loosely-coupled because, in the end, ideally, the CS team's control interface can simply be connected to control the mechanical elements installed in the car.

THE CS TEAM SPECS

The specific aims for the CS team can be divided into several achievement levels:

Level 0 – Minimal useful functionality.

1. <u>An on-board controller.</u> The team will select an appropriate microprocessor (e.g. Raspberry PI, Arduino, etc.) to manage on-board control. The outputs from the microprocessor will be directed to a screen and/or some simple status lights on-board, that indicate the control influences being "applied to the wheels and steering". It is these outputs that would eventually be attached to the physical hardware developed by the ME's; for now, it just needs to indicate the steering/propulsion signals your interface is producing.
2. <u>A control interface.</u> The basic interface for providing control input for the system is an Android app that connects to the on-board controller via Bluetooth. In its most basic version, the mobile app simply presents a virtual joystick on the screen: moving the joystick translates to control signals being sent to the (simulated) steering and traction motors driven by the on-board controller.

Level 1 – A nicely completed application.

1. At this level, features are added to the basic system to make it more complete and usable. Some features to consider include the ability to limit the control actions of the child (e.g., turn off the steering so that only straight ahead travel is possible), emergency stop, limiting a distance or perimeter that the car is allowed to travel; the exact features desirable will be explored with the client.
2. On the car side, the controller will be augmented to accept input from a simple joystick as well. The idea is that this joystick would be mounted to replace the steering wheel, and allow full control for the vehicle.

Level 2 – An exceptional product.

If this "bonus" level is reached, we'd like to explore the true potential of microprocessor control. Once you have control information flowing over Bluetooth, there is no limit to the clever input devices one could develop. One of these might be an "accelerometer vest", that the child wears… and can use to control the car via tilting the body in the desired direction. As a start to exploring this concept, the team will augment the mobile app to use the smartphone's accelerometer: the phone is strapped to a torso; leaning the torso produces appropriate control input.

Of course, these are just overall outlines to help shape a full spec. The team will work with client to refine and detail this spec based on requirements analysis and feasibility study during the fall term. We emphasize the need to keep the design as cheap and easy as possible; ultimately, parents should be able purchase few basic pieces, install your software, and be up and running.

As part of this capstone project, you will be asked to participate in the GoBabyGo club on campus. You are not required to be an officer or have an active role in the club, but must stay current with all builds and events that the club participates in.

Knowledge, Skills, and Expertise you will need to pick up:

Basic knowledge of programming of Auduino, PI, or other chosen system.

Skills in Android programming, including some understanding of Bluetooth communication. Bonus stage involves communicating with the accelerometer.

Basic knowledge of wiring and hardware, enough to plug in a joystick

Equipment Requirements:

- Development machines capable of running Android and PI/Arduino IDEs. Typically any laptop/desktop.
- A Raspberry PI or Arduino or other processor, depending on what is chosen
- A joystick and miscellaneous other small supplies
- An Android smartphone or tablet.

If not available within the team, the sponsor will supply any of the above to the team.

Deliverables:

The deliverables are specifically aimed at making it easy for non-technical parents to install and operate your product. Thus you will need to deliver:

- The smartphone app, ready to simply install on any modern Android device.
- The on-board processor package, delivered as the hardware, plus a software image that can be installed on a storage chip (i.e. microSD), inserted into the device, and you're ready to go.
- As-build documentation giving detailed technical description of how your product works, build specs, and anything else needed by future software teams taking up your project for further development.
- Nice clean pdf manual sufficient to allow any non-technical parent to download, install, and configure your product for successful deployment.