**Final Project Report – Baby Walker 2.0**

**Assistive Robotics Fall 2022**

Alexander Juffras ([juffras.a@northeastern.edu](mailto:juffras.a@northeastern.edu))

Benjamin Fox ([fox.benj@northeastern.edu](mailto:fox.benj@northeastern.edu))

Gabriel Jentis ([jentis.g@northeastern.edu](mailto:jentis.g@northeastern.edu))

Philip Andress (andress.p@northeastern.edu)

Ukhyeon Shin ([shin.uk@northeastern.edu](mailto:shin.uk@northeastern.edu))

William Tower (tower.w@northeastern.edu)

**Abstract**

Baby walkers are commonly used to help the baby move, but at the same time they can also cause various types of injuries. More than 90 percent of the accidents that occur lead to head and neck injuries, and the most common cause of injuries is falling down stairs. Other causes of injury include falling from a walker and proximity-related injuries caused by access to hot spaces such as radiators, fireplaces, or poisonous objects while a child is on a walker in the house. Despite the decline in the number of injuries caused by various efforts, baby walkers still remain a significant and preventable cause of injuries among young children, which supports the American Academy of Pediatrics' request to ban manufacturing and sales in the United States. Therefore, we propose “Baby Walker 2.0” that provides a platform for children to create unique experiences, activities, and entertainment and that is increasingly equipped with safety measures at the same time.

**Background**

A baby walker, or infant walker, consists of a fabric seat with a leg opening and a wheeled base that supports a rigid frame that normally holds the plastic tray. The device supports infants who can mosey with their feet touching the floor and is designed to move while the infant learns to walk. Some walkers are fitted with activity toys, have locking devices that prevent them from moving, and can be folded and stored flat.

Studies have shown that in 2015, the baby walker market size was over $1 billion USD, and 55–92% of 5–15 month infants use walkers, which means mobility is highly preferred by either parents or infants group [1]. There are various reasons for using baby walkers: keeping the baby quiet and happy, encouraging mobility, promoting walking, providing exercise, and holding the baby during feeding. According to *Pediatrics’s* study [2], one-third of the parents surveyed said they used walkers because they believed they would keep their babies safe. However, in 1999, about 8,800 children under 15 months of age in the United States visited hospital emergency rooms for injuries related to baby walker use [3]. In addition, the National Electronic Injury Surveillance System (NEISS) reported 230,676 children under 15 months of age who were treated for walker-related injuries in the emergency room from 1990 to 2014, most of which occurred on the stairs at home, as shown in the table below. [[4]](https://www.cureus.com/articles/63047-the-reasons-and-associated-injuries-related-to-baby-walkers-use-among-children-in-riyadh-saudi-arabia#references).

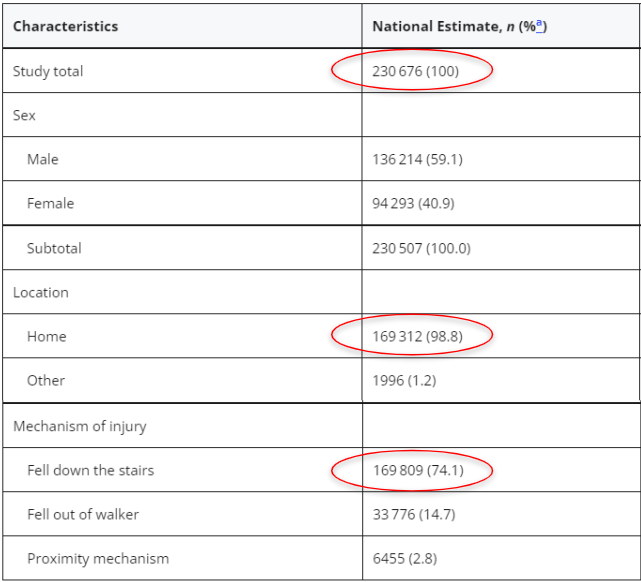


Table 1. Characteristics of Infant Walker–Related Injuries Among Children <15 Months Old Treated in US EDs (NEISS 1990–2014) [4]

Approximately a quarter of baby walker-related injuries reported to NEISS are described as "more serious," and almost all of which are fracture and obstructive head injuries. Skull fractures accounted for nearly 10% of all baby walker-related injuries in one large patient series.

Fig 1 shows the mechanism of pedestrian-related injuries with data from other research institutes. Similarly, injuries have been shown to be mostly caused by falls from the walker or with the infant remaining in the walker, and stairs are associated with 75% to 96% of cases and almost all serious injuries [5]. In addition, a small number of pinch injuries occur on fingers and toes, and burns account for 2 –5% of walking-related injuries. Walkers are also generally associated with poisonous materials in infants under the age of one [6]. This burn and toxicant contact appears to be due to increased access to these hazards as infants increase mobility in the walker.

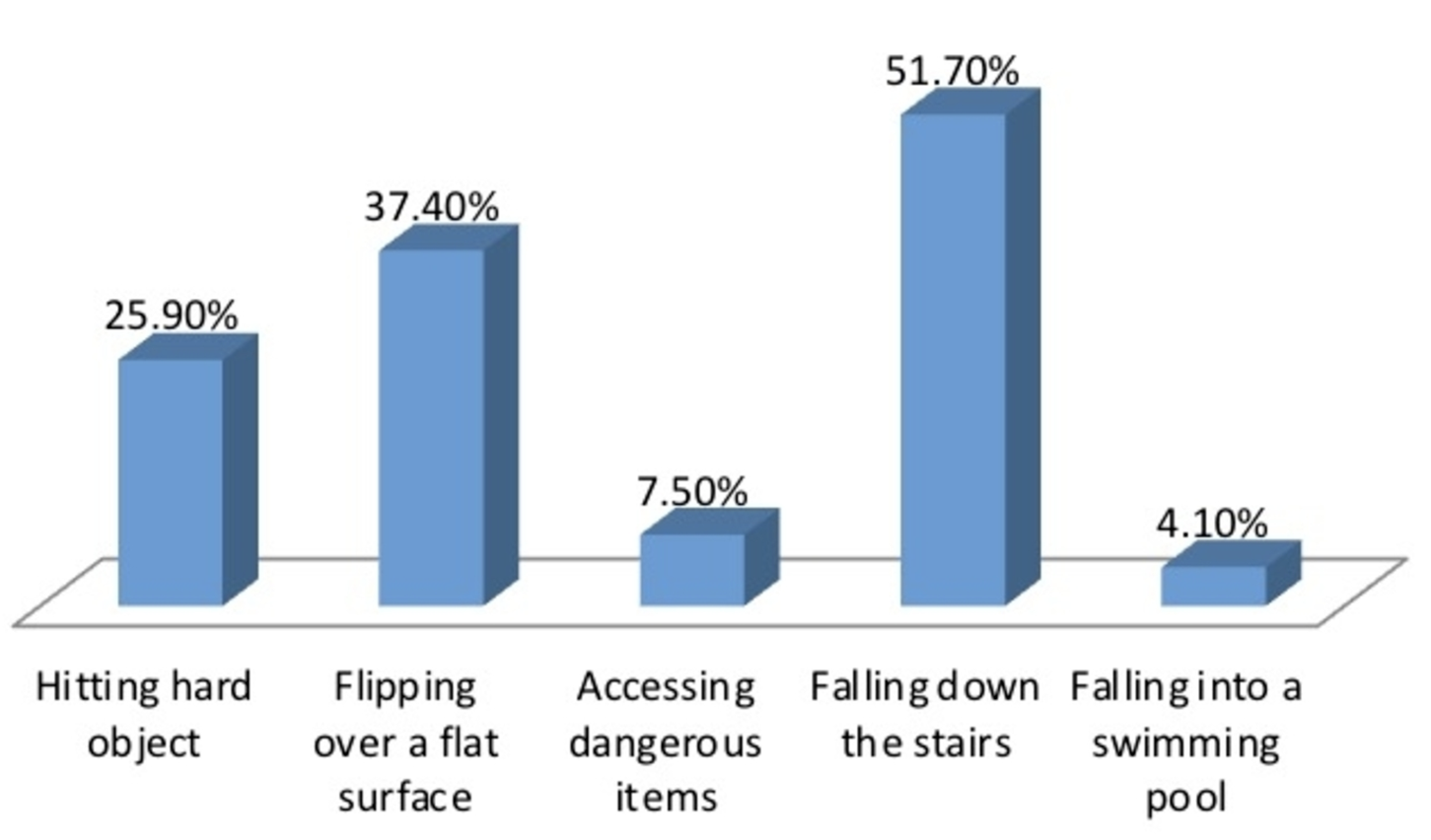


Fig 1. Mechanism of injury while using baby walkers [7]

As a result, certain voluntary safety standards have been established, and organizations and governments have begun to advocate or implement policies targeting walkers to protect children. For example, on April 7, 2004, Canada banned the import and sale of baby walkers [8]. The American Academy of Pediatrics not only proposes a ban on the production and sale of mobile walkers, but also advocates parents abandoning walkers at home and entertaining their children in fixed activity centers and other activities [3]. In 2010, the U.S. Consumer Product Safety Commission mandated and made these standards stricter. such as the requirement that "Child walker frames should be wider than standard 36-inch entrances or that there should be braking if more than one wheel falls on the edge of the stairs” [9].

While changes in safety standards have certainly led to a steep drop in the number of babies injured using infant walkers, more than 9,000 US children are still injured using the devices every year [10]. Most baby walker-related accidents happen even when the baby is with a caregiver, since a child in a walker can move more than 3 feet in 1 second, which is beyond a parent's expectation. Thus, even with an adult close by, walkers are not safe to use [11].

**Competitive Landscape**

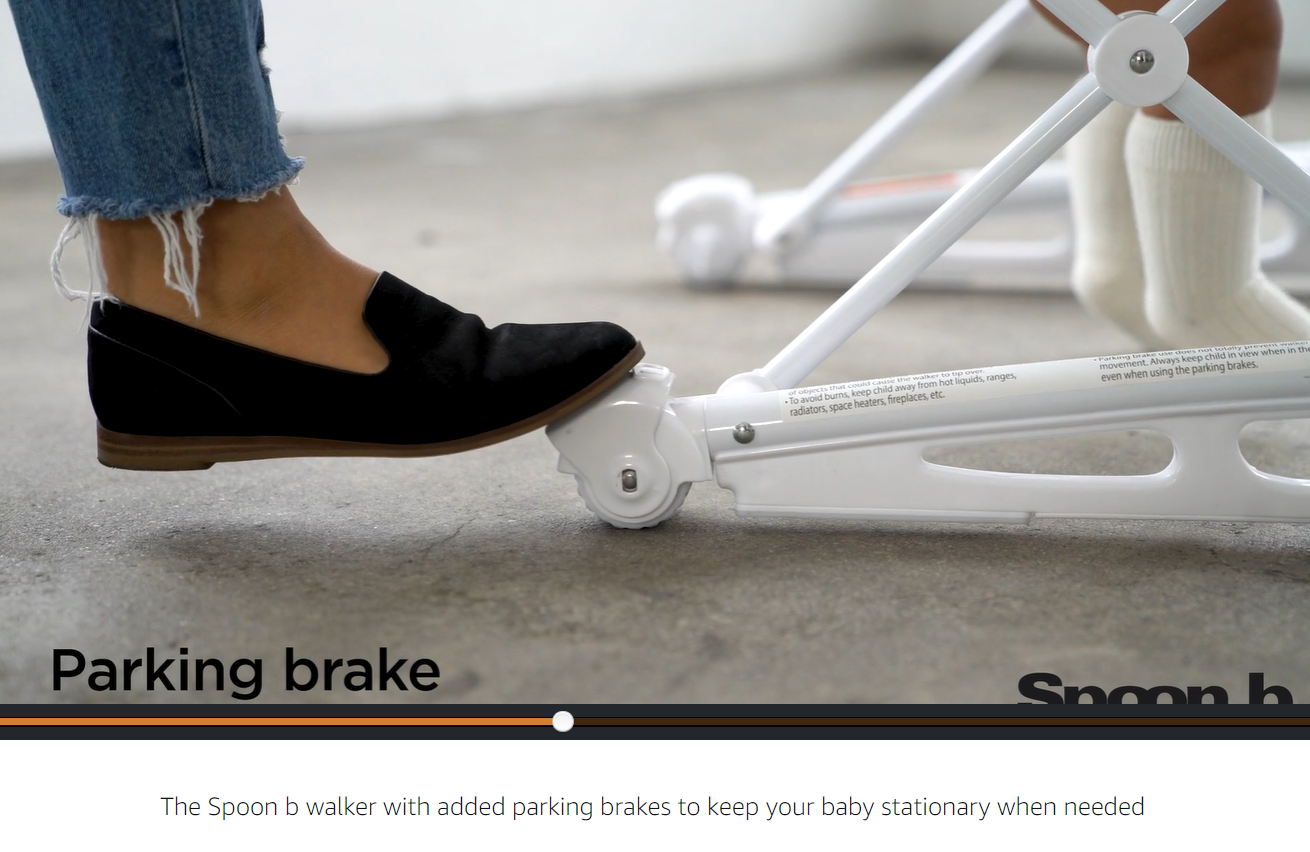
****

Fig 2. Ad for manual brakes in baby walker [12]

Fig 2 shows a scene from an advertisement from the baby walker company Joovy for the Spoon B that is loved by many parents. This company simply installed handbrakes on the rear wheels. Despite the fact that the price is more than twice that of regular products, parents are happy to buy it [12]. However, the downsides of this product are that it still requires parental supervision at all times and that the baby's freedom is completely dependent on the parents’ decision.

**Introduction of Baby Walker 2.0**

To solve the above problem, we made an assistive device that provides the maximum autonomous movement for babies to create their own unique experiences, activities, and entertainment while ensuring the safety of babies as a top priority. Therefore, we designed a ‘Baby Walker 2.0’ that allows the baby to move and play freely within the safety line set by parents, but automatically stops the wheels when the baby tries to cross the safety line.

**HAAT**

• **Human:** There are two users with respect to this robot. The baby wants to learn more about the world and how it works. The parent wants to keep the little one safe. Supervision and safety precautions are the number one priority for them.

• **Activity:** Walking and exploring with a baby walker.

• **Assistive Technology:** Device to sense a parent-defined boundary that prevents the baby from entering dangerous areas.

• **Context:** Walking and exploring with a baby walker. Protecting babies is important if you want to let them roam, so parental supervision and gates exist as solutions. However, it is tiring and a lot of effort for parents to be constantly present. What about protecting babies with the walker itself?

**Solution Behavior**

The baby walker requires a boundary that the baby moves around in, defined by magnetic strips on the floor. When one of the wheels exits the boundary, the opposite wheel locks to let the baby move back into the boundary. If the baby finds a way to exit the region, signified by two wheels crossing the boundary, the baby walker will lock all corners and alert the parent via buzzer. The magnet strip should be placed far enough away from the element of danger that if two wheels pass the magnet strip, the baby will not be in a threatening situation. The logic flowchart is shown below.

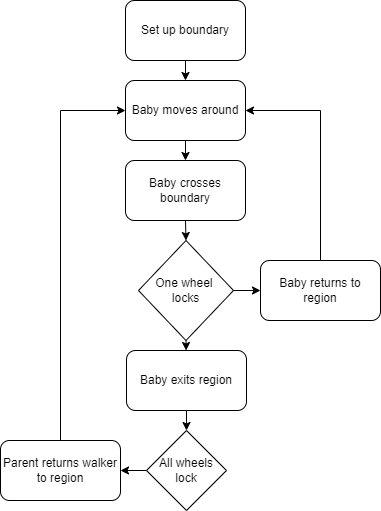


Fig 3. Walker 2.0 behavior Flow Chart

**Approach and Implementation**

**Hardware**

*Task Breakdown*

* **Philip:** Mechanical & Electrical Design
* **Alex:** Electrical Design

*Mechanical Design*

For the brakes, we initially planned on using cable actuated disk brakes commonly found on RC cars. However, upon considering that baby walkers generally have a mix of casters and fixed wheels that are made of flimsy plastic, we opted to bypass the wheels altogether by using linear actuators to apply force directly to the ground. This choice simplified the design greatly and reduced the mechanical workload to designing and 3D printing a stable mounting point to connect the linear actuators and hall-effect sensors to the frame of the baby walker.

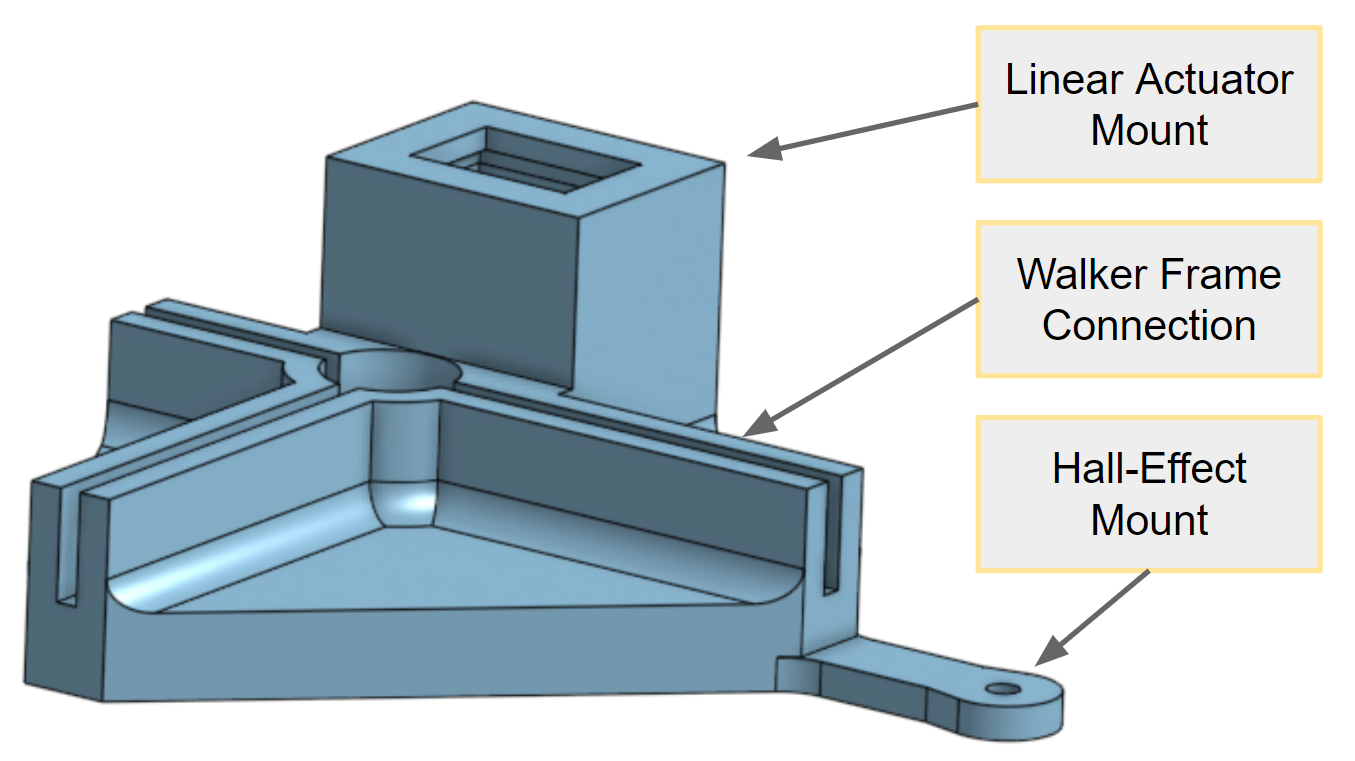


Fig 4. Brake Mount CAD

In order to make this system as unobtrusive as possible, we designed it with a friction fit to the existing support structure built into the baby walker’s frame. The same design with slight alterations was replicated for each of the four corners of the baby walker.

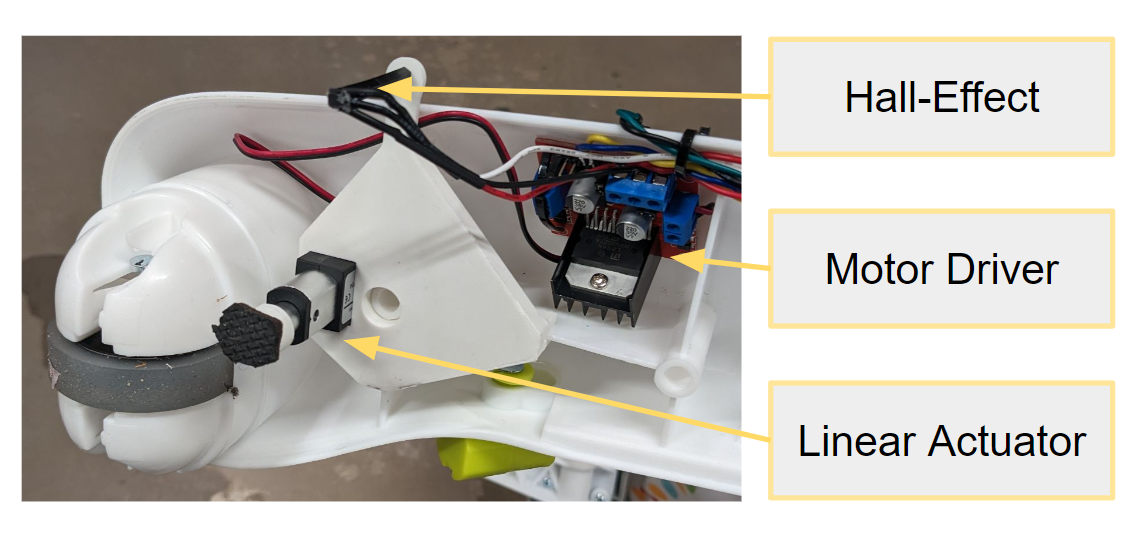


Fig 5. Integration with Baby Walker

*Electrical Design*

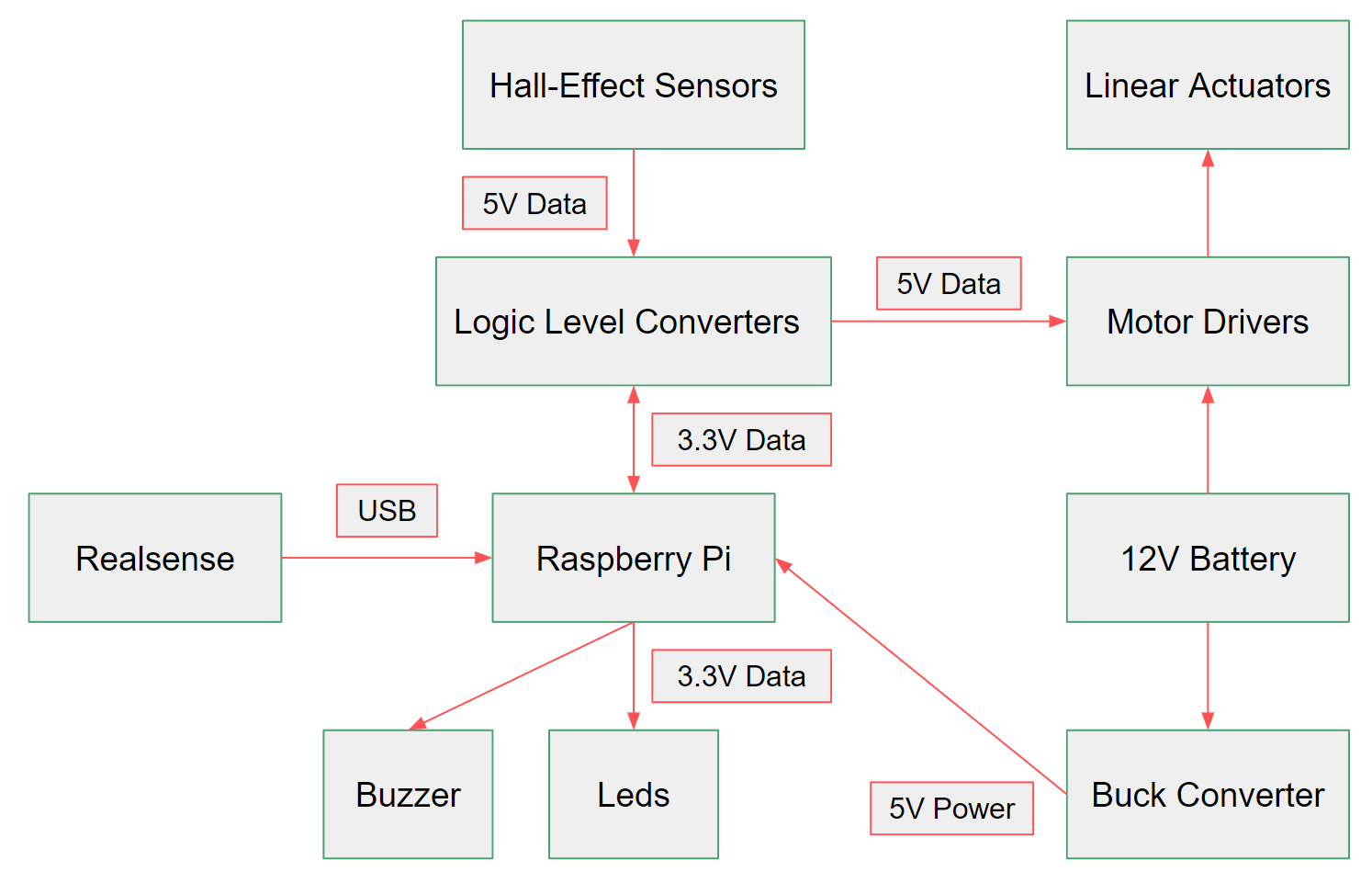


Fig 6. System Electrical Block Diagram

Our system is powered by a 12V LiPo battery. The battery is connected directly to the motor divers and to a buck converter that steps the voltage down to 5V then via USB to Micro USB to the Raspberry Pi. The 5V line is connected to the Raspberry Pi, logic level converts, motor drivers, and the hall-effect sensors. The logic level converters were wired on a protoboard and screw terminal blocks were used to make connections. The data lines of the hall-effect sensors and motor drivers are both routed through the logic level converters in order to make the signal voltage compatible with the Pi’s GPIO pins. As the leds and buzzer can operate with 3.3V we opted to bypass the logic level converters and route them directly to the Pi. Finally, the Intel Realsense is connected to the Pi via USB.

**Software**

*Task Breakdown*

* **Benjamin**: RPi Setup, Menu & Servo Drivers
* **Gabriel**: Main Logic, Hall Effect & Buzzer Drivers
* **William:** Intel RealSense

*Software Architecture*

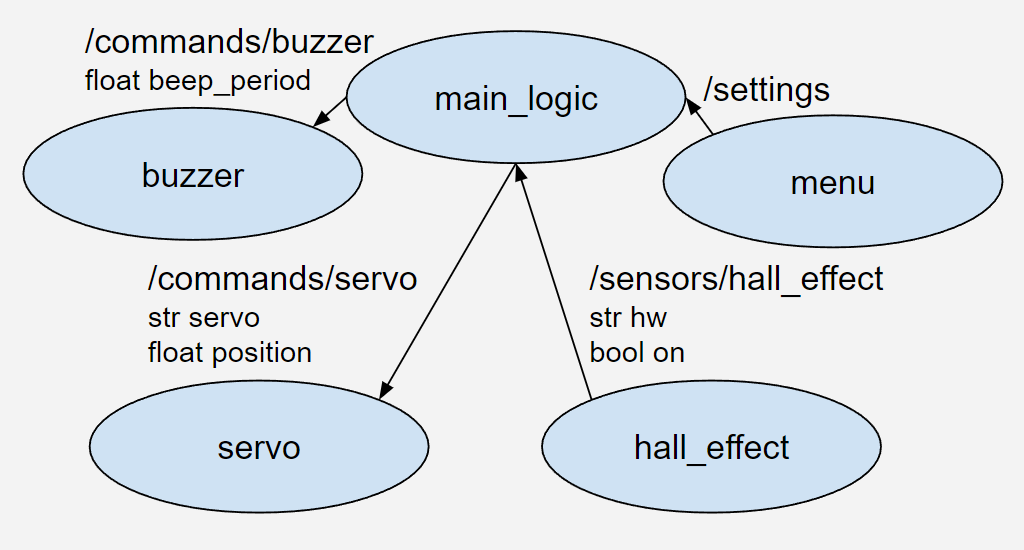


Fig 7. System Software Block Diagram

The software was planned to run in a ROS environment. This allowed parallel software development of drivers and logic, with only the necessary knowledge being nodes and messages. Therefore, the design had to be conceived first. The Intel RealSense is left out since the RealSense was not properly integrated into the system, which will be further explained in a later section.

The software consisted of four hardware drivers. Their hardware pins were exposed as parameters in the form of a ROS launch file. At a high level, the buzzer node needed a period to beep at, or 0 to be off. The servo node could be given a servo (front/back left/right) and a target position. The hall effect node just needed to notify when the hall turned on or off, so the message consisted of the hall effect location on the walker, plus the on flag. The menu published a set of subtopics under “settings.” One topic per setting was used, along with a value for the setting. Finally, due to the relative simplicity of the software, a single node was sufficient to hold all the logic from the drivers.

*Raspberry Pi ROS Setup*

Since the Raspberry Pi was a slightly older 3B+, an older Raspbian operating system had to be installed on it: Buster. Buster was more consistently compatible with older versions of ROS1, so ROS Melodic was chosen. The instructions in [14] were used to set up the ROS workspace on the Pi. These instructions built ROS packages from source, using *catkin\_make\_isolated*. This is a more flexible and industry-consistent approach to using ROS, compared to the *catkin\_make* build approach used in many tutorials [15]. To develop on the Pi, a README was created, and an ethernet-USB converter was provided for members without an ethernet port.

*Drivers*

**Menu Driver**: The menu driver consisted of buttons and a menu class. The menu class had two levels of menus: the top-level settings and lower-level values. Up and down triggers would allow the current sub-menu to scroll between options. The right trigger would go to the submenu, while the left trigger would lock in the current option and return to the top menu. Upon locking, the setting that was changed was published to its appropriate topic. The topics in the created menu were {enter new environment: yes, no} and {brake power: 1-5}. A new environment would reset the RealSense SLAM if integrated, and the brake power would modify the braked position of the linear actuator.

The menu driver was provided four button pin numbers as input, and it set them up as digital inputs. Each digital input rising edge was set to trigger the appropriate button in the menu class. In the end prototype, the menu was not implemented in hardware physically due to an oversight not ordering an LCD screen. However, testing with buttons and print statements worked as expected.

**Servo Driver**: The servo driver was designed to control the position of the linear actuator. Since the brakes were supposed to act as fast as possible, speed control wasn’t important. The brakes should reach the target position as fast as possible. Therefore, PWM speed control of the actuators weren’t necessary, and each linear actuator could have a 100% duty cycle. This is the same as controlling with digital pins. On node start, it took in eight pins as inputs: two per linear actuator. All pins were set up as digital outputs, and the behavior table for each actuator as a function of the two inputs is shown below.

| Behavior | Forward | Backward | Stationary |
| --- | --- | --- | --- |
| Input 1 | High | Low | Low |
| Input 2 | Low | High | Low |

TABLE 2: Walker Braking Logic

The servo driver also tracked the assumed position of each linear actuator, initializing to retracted at 0. The position tracking was open-loop due to the lack of need for precise position control. If the brake didn’t perfectly extend, it wouldn’t affect the performance. The driver subscribed to a topic whose messages contained the servo (“fr”, “fl”, “bl”, or “br” for front/back left/right) and target position. Upon seeing the message, the appropriate actuator would be turned on in the appropriate direction. The updated position was set at this time, too. Then, a one-time timer was started for the actuator’s time of travel. The timer end trigger then turned off the actuator.

**Buzzer Driver**: The buzzer driver was written for a single buzzer. It used a predetermined PWM frequency that controlled the pitch of the sound made by the buzzer. The goal was to have the buzzer start making noise when the baby walker was fully locked by the brake system. This would provide additional feedback to the caretakers to reset the baby walker in the safe range. This driver received data from a buzzer message published by the main logic node. This message contained a delay value to delay the time between tones played by the buzzer. The buzzer stopped making sound when it received a message with a delay time of 0 seconds.

**Hall Effect Driver**: The hall effect driver handles the reception of the signals from the four hall effect sensors positioned near each wheel of the baby walker. The driver waited for one of the hall effect sensors to detect the magnetic strips on the ground aimed at preventing the baby from interacting with a hazardous environment. The driver looked for a change in detection, meaning either detecting a magnet when not previously detecting one, or detecting no magnet after detecting one. Using the rising and falling edge signals, the driver published messages to a topic intended for the main logic. These messages consisted of two parts: a string representing the location of the sensor, (using the same representation as mentioned in the servo driver section) and a boolean value for whether the sensor started or stopped detecting a magnet.

*Main Logic*

The main logic first initialized connections to the other nodes in the system. It created subscribers to get data from the menu driver and hall effect driver. Had the Intel RealSense been integrated, it would have been given a subscriber for data, too. This will be expanded upon in the next section. The node then also created a publisher for the servos and buzzers.

To control the servos, the node first got the data about the desired brake power for the system. The brake power was mapped to an extended position of the servo. It then waited for a message from the hall effect sensors for a detection that the baby walker had approached a hazardous region. If it had only received a message about a single wheel crossing, a message was sent to the servo topic to apply the opposite wheel’s brake (i.e., if the front left sensor has a detection, the back right servo is told to actuate). This was done to allow the baby to maneuver the walker back into the safe region. If the walker then returned to the designated safe area in the environment, a message was sent to retract that servo, granting full movement to the baby walker. Had multiple sensors sent messages about detecting the magnetic strip, the main node sent a message to all servos to actuate and fully brake the walker. It also sent a message to the buzzer so as to activate an audio cue that the walker was in a dangerous position. When the baby walker was returned to the safe zone and designated that way by the parent via the new environment menu setting, the logic sent messages to retract the brakes on all the wheels and turn off the buzzer.

For the prototype and testing of this system, this main logic was modified slightly due to the presence of only a single linear actuator brake. Only a single wheel was set up to have sensing and braking capabilities. Therefore, the response to the detection of the hall effect sensor was modified. In this implementation, when the hall effect driver published the message for the detection of the magnetic strip, the system treated it as a full exit of the safe region, sending a message to activate the brake on the single corner and turn on the buzzer.

*Intel RealSense*

Simultaneous Localization and Mapping (SLAM), is a technique for building a map of the environment and determining location within that map at the same time. SLAM methods are commonly used for navigation and localization tasks. This is especially relevant in many robotics applications where systems:

a. are often in novel environments

b. need to know where they are

The baby walker is one such system. First, a high quality map in an arbitrary room is convenient for the caregiver, which can lead to freer exploration for the child. Second, high quality localization is critical for ensuring the walker remains in a safe state. It follows that SLAM methods are a natural choice for the walker.

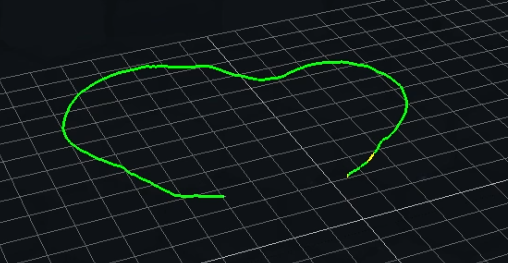


Fig 8. Odometry from VI-SLAM with the RS-T265

The Intel RealSense T265 camera is an off the shelf product commonly used to resolve vi-slam, which relies on landmark visual data for longer-term stable localization and mapping, and IMU data for moment to moment variation. SLAM capabilities for dynamic mapping were implemented with pose estimation, landmark localization, and loop closure on a standalone PC. This implementation was aided greatly by Intel's C++ library for working with the device, librealsense2, and by Intel's device relevant example ROS modules, which make nuts-and-bolts details clear [13].

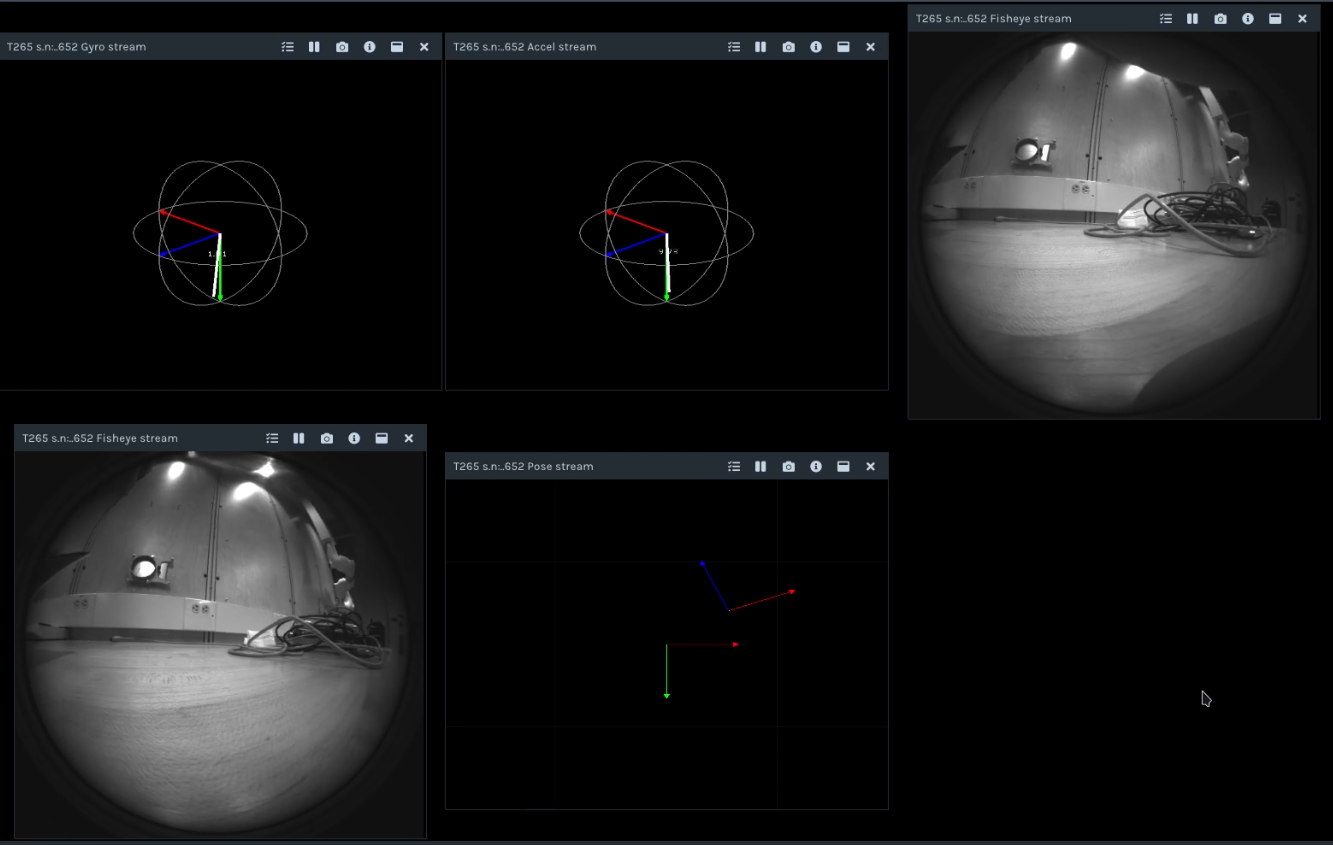


Fig 9. Streamed Sensor Data from the RS-T265

Several prior design choices made integrating the RealSense into the walker very difficult, such that it could not be accomplished in the scope of this project. Two stand out:

a. Due to the school’s security restrictions on connecting embedded devices to their network, the lack of wireless SSH access to the Raspberry Pi led to slower iteration time on code and imposed a location constraint on development. Compounding this, SLAM implementation fell to an individual without an ethernet port.

b. The choice to use ROS Melodic was made early on in the project, but this version of ROS proved to be quite difficult to work with. Crossed wires between subteams led to the Intel Real-Sense development on ROS Noetic for the majority of the project. Because main module ROS nodes were being developed on the Pi, Real-Sense capabilities were developed separately, and so the discrepancy was only discovered very late in the build cycle. When the team began module integration the build system was unfamiliar. Only catkin\_make\_isolated (not catkin\_make) was supported on the Pi’s ROS installation, and the permissioning and utility installation structures led to significant difficulty building and linking the libraries necessary for resolving SLAM on the T265. To complicate matters, other project ROS modules had been built and configured to work only under root user access, and several core utilities had been configured without default user access. There were significant negative impacts on integration timelines as a result.

**Results**

In the final integrated prototype system, the hardware consisted of 1 corner containing a mounted hall effect sensor and a brake. As described earlier, this was done due to having possession of a single linear actuator. Demonstrating the system with a recorded video with only 1 brake would be easier with both the sensor and brake in the same corner.

Additionally, a buzzer was attached, and these pieces of hardware connected to the Raspberry Pi for signal and the battery for power through the appropriate voltage converters. This system is shown in the pictures below.

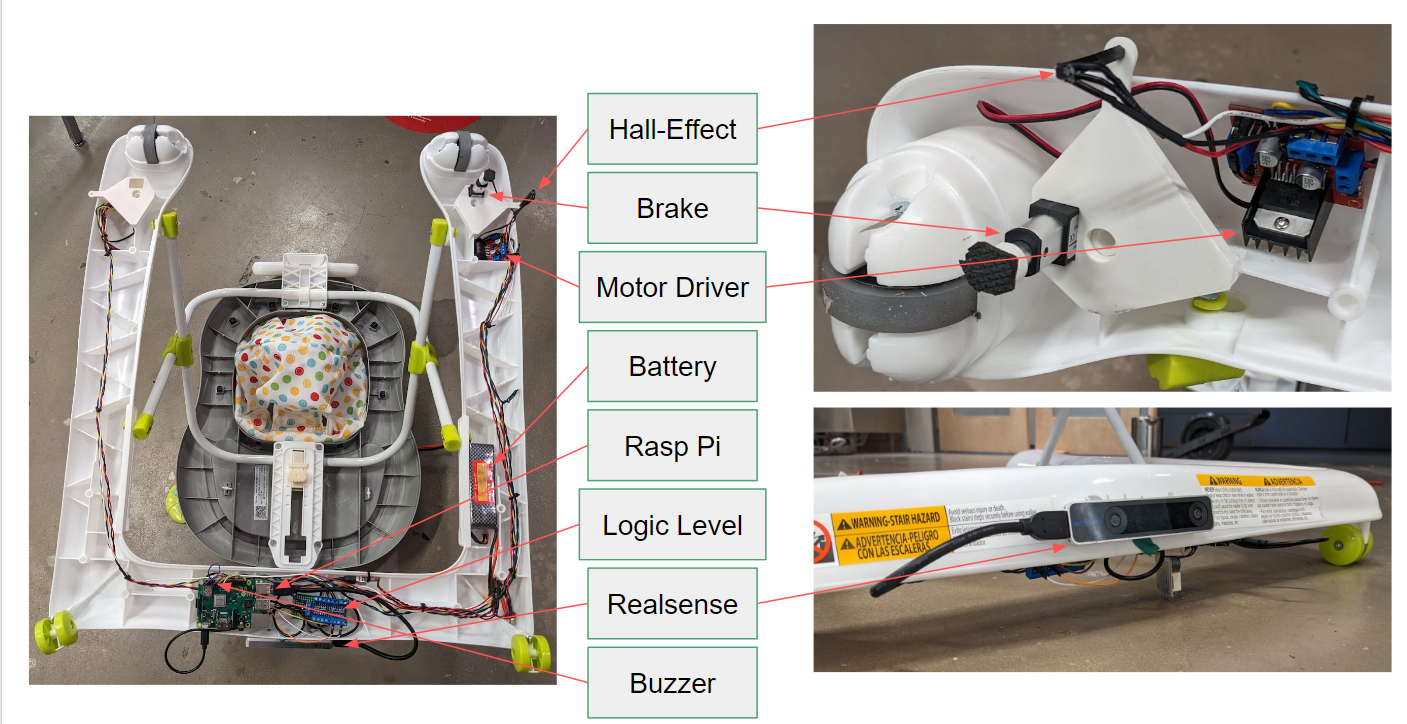


Fig 10. Final Electrical Layout of Baby Walker 2.0

****

Fig 11. Baby Walker 2.0

When the hall effect passed over a magnetic strip on the ground, the brake successfully lifted, as expected. This is shown in the successive photos below. The buzzer also went off for demonstration purposes, but this would normally only happen if two wheels crossed the boundary.



Fig 12. (left) Hall effect sensor detecting magnetic strip (right) linear actuator braking baby walker

As shown in the presentation video, the other three wheels were able to rotate about the extended linear actuator. Typically, since this would be in the opposite corner, the turning radius relative to the boundary would look like the following image. Note the magnetic strip won’t usually be tangent to the turning radius.

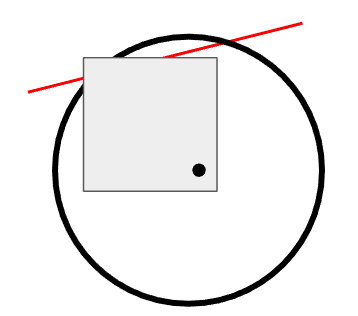


Fig 13. Turning radius of baby walker relative to boundary

Due to RealSense integration trouble and the inability to brake multiple wheels, full expected system behavior couldn’t be tested. In theory, after a single wheel crossed the strip, the RealSense would have been used to determine whether the walker continued beyond the boundary or stayed within. If a second wheel crossed the boundary while the RealSense thought the system was outside the boundary, then the full locking of four brakes would have been activated.

**Conclusion**

Infant walker–related injuries decreased after the implementation of the federal mandatory safety standard in 2010. This decrease may, in part, be attributable to the standard as well as other factors, such as decreased infant walker use and fewer older infant walkers in homes. Despite the decline in injuries, infant walkers remain an important and preventable source of injury among young children, which supports the American Academy of Pediatrics’ call for a ban on their manufacture and sale in the United States.

With this innovative device, parents will be able to use a baby walker that would limit dangerous situations where a baby would be hurt. With an opportunity to further expand on the development of the safety features, this project could progress into more a potential commercial product. Pending government review, this device could let parents use baby walkers in countries where there are bans on baby walkers.

**Work Cited**

[1] “Global Baby Walker (Sit in & Standing) Market Outlook, 2026” *Bonafide research*, Aug, 2021. [Online serial]. Available: <https://www.bonafideresearch.com/product/210819721/global-baby-walker-market>

[2] Injuries Associated With Infant Walkers, *Pediatrics* (1995) 95 (5): 778–780.

<https://doi.org/10.1542/peds.95.5.778>

[3] American Academy of Pediatrics. Committee on Injury and Poison Prevention: [Injuries associated with infant walkers](https://dx.doi.org/10.1542/peds.108.3.790?utm_medium=email&utm_source=transaction). *Pediatrics*. 2001, 108:790-2. [10.1542/peds.108.3.790](https://dx.doi.org/10.1542/peds.108.3.790?utm_medium=email&utm_source=transaction)

[4] Sims A, Chounthirath T, Yang J, Hodges NL, Smith GA: [Infant walker-related injuries in the United States](https://dx.doi.org/10.1542/peds.2017-4332?utm_medium=email&utm_source=transaction). Pediatrics. 2018, 142:[10.1542/peds.2017-4332](https://dx.doi.org/10.1542/peds.2017-4332?utm_medium=email&utm_source=transaction)

[5] MDMichael D., [MD Jill A](https://www.annemergmed.com/article/S0196-0644(05)82386-8/pdf#)., [MD Fredric B.](https://www.annemergmed.com/article/S0196-0644(05)82386-8/pdf#) Head injury and the use of baby walkers: A continuing problem, *Annals of Emergency Medicine; an annual journal* <https://doi.org/10.1016/S0196-0644(05)82386-8>

[6] [Genevieve H](https://www.mamanatural.com/about/)., Baby Walkers: Why Canada Banned This Popular Registry Item, *mamanatural,* July 06, 2021

https://www.mamanatural.com/baby-walkers/

[7] Albarrak Y A, Algwaiz A F, Sharaf A M, et al. (August 12, 2021) The Reasons and Associated Injuries Related to Baby Walkers Use Among Children in Riyadh, Saudi Arabia. *Cureus* 13(8): e17122. doi:[10.7759/cureus.17122](https://www.cureus.com/articles/63047-the-reasons-and-associated-injuries-related-to-baby-walkers-use-among-children-in-riyadh-saudi-arabia)

[8] [Information for Canadians travelling outside of Canada](https://www.canada.ca/en/health-canada/services/consumer-product-safety/advisories-warnings-recalls/information-canadians-travelling-outside-canada.html?utm_medium=email&utm_source=transaction). (2020). *Government of Canada* [https://www.canada.ca/en/health-canada/services/consumer-product-safety/advisories-warnings-recalls/information-canad...](https://www.canada.ca/en/health-canada/services/consumer-product-safety/advisories-warnings-recalls/information-canadians-travelling-outside-canada.html?utm_medium=email&utm_source=transaction).

[9] Robert J. May 12, 2010, [Consumer Product Safety Improvement Act of 2008 (CPSIA), Draft Final Rule for a Safety Standard for Infant Walkers](https://www.cpsc.gov/s3fs-public/pdfs/blk_pdf_walkers.pdf), *UNITED STATES CONSUMER PRODUCT SAFETY COMMISSION,* <https://www.cpsc.gov/s3fs-public/pdfs/blk_pdf_walkers.pdf>

[10] Naomi T., September 17, 2018, More than 9,000 US children are injured by infant walkers every year, study finds. *CNN.*

https://www.cnn.com/2018/09/17/health/infant-walkers-injury-parenting/index.html

[11] American Academy of Pediatrics, Aug 15, 2022,

Baby Walkers: A Dangerous Choice, *healthy children*

https://www.healthychildren.org/English/safety-prevention/at-home/Pages/Baby-Walkers-A-Dangerous-Choice.aspx

[12] Spoon-b, Joovy, <https://joovy.com/spoon-b/>

[13] Intel, "Intel RealSense Tracking Camera," Document Number: 572522-004, Jan. 2019 [Revised Sept. 2019]

[14] “Installing ROS Melodic on the Raspberry Pi,” *ROS.org.* <http://wiki.ros.org/ROSberryPi/Installing%20ROS%20Melodic%20on%20the%20Raspberry%20Pi>

[15] “catkin\_make\_isolated for non-standard catkin packages,” *ROS.org*. <https://ros.org/reps/rep-0134.html>