**HARDWARE RESEARCH**

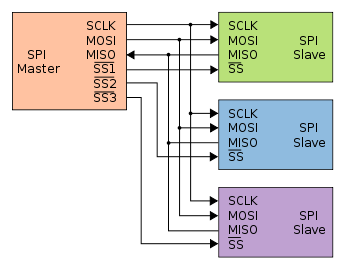
# **ARDUINO AND RFID**

An Arduino device, which is our current solution for a way to handle and process RFID inputs, cannot run a Unity application. We should still consider purchasing one to handle RFID processing.

It seems like Unity *could* build for a Raspberry Pi; a Raspberry Pi could also be used to handle the RFID input. However, I think it’s best to have and use two separate devices for handling the RFID and running the Unity application. This would let us swap in and out the device running the Unity application. We would want this because there are a number of caveats that come with trying to build a Unity application for a Raspberry Pi—critically, Unity basically only supports x86 builds.

A Raspberry Pi can run Ubuntu Linux, but it uses an ARM ISA, so the build could have problems. As our application shouldn’t be incredibly demanding, it could be the case that this is not a problem at all. There are also workarounds involving building the application for Android (in this case the Pi would be using an Android OS) or WebGL, but they don’t seem like surefire approaches; should we pursue them and fail, the application could also be built to run on an external x86 device (i.e. an x86 laptop or a UP-7000) that outputs to the display in the table.

It seems like we should use an [Arduino Mega](https://store-usa.arduino.cc/products/arduino-mega-2560-rev3?selectedStore=us) due to its greater number of pins compared to the Arduino Uno. The [RC522 RFID Reader](https://www.amazon.com/HiLetgo-RFID-Kit-Arduino-Raspberry/dp/B01CSTW0IA)[[1]](#footnote-1) appears to be the only/main/most popular option. I think that it uses the [Serial Peripheral Interface (SPI) to connect to the Arduino](https://www.youtube.com/watch?v=ZGaCXHvgcE4), which uses 4 lines to connect. Each SPI device needs its own SS line, but I believe that the MOSI and MISO lines can be shared, and the clock definitely can. We should increase the number of RFID scanners in the device from 5 to 10; this will allow children the space to experiment with concepts such as verb tenses. [RFID tags are cheap and easy to source](https://www.digikey.com/en/products/detail/texas-instruments/RF37S114HTFJB/5798060?utm_adgroup=&utm_source=google&utm_medium=cpc&utm_campaign=PMax%20Shopping_Product_Medium%20ROAS%20Categories&utm_term=&utm_content=&utm_id=go_cmp-20223376311_adg-_ad-__dev-c_ext-_prd-5798060_sig-Cj0KCQiAoKeuBhCoARIsAB4WxtcOwsY7ZgWgyKtTyaWgP4526oQV_mYQ51-JG1HSmbvQoDB3IllE2zEaAiLsEALw_wcB&gad_source=1&gclid=Cj0KCQiAoKeuBhCoARIsAB4WxtcOwsY7ZgWgyKtTyaWgP4526oQV_mYQ51-JG1HSmbvQoDB3IllE2zEaAiLsEALw_wcB)[[2]](#footnote-2); we can secure 30, which gives us enough for every letter, one “clear space” block, and 3 extras.



# **RASPBERRY PI/ALTERNATIVES AND DISPLAY**

As I mentioned above, a Raspberry Pi could run a Unity application in theory. It’s possible that our application does not require incredible performance and runs fine despite not being a native ARM build; it’s also possible to try and build for Android or WebGL. In any case, we should consider the Raspberry Pi 4 or older, as the new Raspberry Pi 5 is more expensive and we would likely not come close to full utilization.

There are some x86 single board computers as well, such as the [UP 7000](https://www.cnx-software.com/2023/07/26/up-7000-powerful-intel-alder-lake-n-alternative-raspberry-pi-4-sbc/); however, these are not cheap. We would be better served running the application on a device behind the product that is connected to the desk’s screen for output. Both boards, and obviously a computer, can connect to the internet—we will need this for our API calls.

I am not sure how to find a display that could be set into the desk (i.e. a display that isn’t a monitor). I am confident that this would not be particularly expensive relative to the rest of the materials. The display size should be no less than 10 inches; a 13.3-inch display would be ideal.

1. It’s even cheaper on TEMU. [↑](#footnote-ref-1)
2. These *seem* like they should work, but I am learning about all this in real time. I could be wrong. [↑](#footnote-ref-2)