E)

1) By separating the display drivers from our fixed.c subroutines our fixed.c code becomes modular. This means we can easily call our fixed.c subroutines from future projects without worrying that display drivers functions are being executed.

2) If the decimal does not remain in the same physical place then there is no alignment of the output values to the display. This makes reading information posted to the screen difficult. Additionally, in our display drivers we know the output should be 6 ascii value characters, if there is no alignment we must ensure that no additional/incorrect values are displayed to the screen. For example, a requested output of 0.00 must end after displaying the ‘0.00’ and not continue displaying 6 characters.

3) Fixed-point over floating point: if the application requires high speed, fixed range of values, and moderate precision. Floating point over fixed: If the application requires high precision, has a large unbounded range of values, and speed is not as large of an issue.

4) Binary Fixed point over Fixed point: If there is a bounded range of values, and there is a need for very high speed (can be accomplished with left shifts instead of division).

Fixed over binary: Manual verification of overflow and dropout, writing the code is not as easy to decipher from the human eye. Not many benefits over fixed point

5) Example Application: 2 Sig Fig Household Thermostat would not need floating point because precision past 2 decimal points for temperature is irrelevant for household use. Additionally, by avoiding floating point we can use a cheaper microprocessor and consume less power. We would want to capture the temperature from a temp module and convert that number into a 2 decimal place with 2 sig figs IE(92.72\* Degrees F) string.

6) Yes we can use floating point on the ARM M4, there is separate floating point hardware. Our cost is computation time, more power consumption by using additional hardware, more complex code to interface with the floating point module.